Extraction d'attributs et mesures de similarité basées sur la forme

Mingqiang Yang

N° d'Ordre:

UMR 6164

Ecole Doctorale MATISSE

THESE

présentée à

L'INSTITUT NATIONAL DES SCIENCES APPLIQUEES

pour l'obtention du

DOCTORAT

Spécialité : **Traitement du signal et de l'image**

par

YANG Mingqiang

Shape-based feature extraction and similarity matching

Extraction d'attributs et mesures de similarité basées sur la forme

---

Soutenue le 3 Juillet 2008

Directeur de Thèse: **M. RONSIN Joseph**
Co-encadrant: **M. KPALMA Kidiyo**

Membres du jury :

COQUIN Didier	**Maître de conférences (HDR) à Université de Savoie**
JURIE Frédéric	**Professeur à Université de Caen**
KPALMA Kidiyo	**Maître de Conférences à l'INSA de Rennes**
MAITRE Henri	**Professeur à TELECOM Paris**
RONSIN Joseph	**Professeur à l'INSA de Rennes**

To my parents

To my beloved wife and daughter

To all my friends

ACKNOWLEDGEMENT

I'm very lucky to have the chance to study the shape-based feature extraction and similarity measures:
This research is a challenging task, because it is the fusion of many subjects in different fields.
This research is prospective domain, because it can be applied to many aspects in modern life.
This research is worthy of research, because it is not perfect by far.

When I sit in front of my excellent computer, read the tremendous amount of relevant literatures, test the new methods by experiments, learn the knowledge and technology from my supervisors, discuss some new approaches with my colleagues, the great joy fills in my heart.

Now, it is the time for a last and personal word. During those 2 years and 10 months in Institut National des Sciences Appliquées (INSA) of RENNES, I have met a lot of people outside and inside the work sphere that contributed to make this adventure possible and enjoyable. So the purpose of this page is to say "thank you" to all of you.

First of all I would like to begin with my two supervisors, Maître de Conférences, Dr. Kidiyo KPALMA and Prof. Joseph RONSIN, thank you for giving me the opportunity to develop this experience abroad and challenging me every day to make me a better researcher. Your writing skills and diligence for the corrections of my manuscript helped relieve many writing blocks. It was always a pleasure to share with you new results and your help, stimulating suggestions, constant cheering, interest, enthusiasm and encouragement helped me in all the time of research to push through and complete this thesis.

Furthermore, I would like to express my gratitude to all those who gave me the possibility to complete this thesis. I want to thank the Groupe Image et Télédétection in Institut d'Electronique et de Télécommunications de Rennes (IETR) for giving me permission to commence this thesis, to do the necessary research work and to use departmental devices and data. My former colleagues from the School of Information Science and Technology in Shandong University, China, supported me in my research work. Especially I am obliged to Dean Dongfeng Yuan and Prof. Yuhua Peng. I want to thank them for all their help, support, interest and valuable hints.

I am grateful for the scholarship provided by China Scholarship Council (CSC). Without this help I could not complete this field research.

Finally, I would like to give my special thanks to my wife Hong and my daughter Jiajie whose great moral support and patient love enabled me to complete this work.

Contents

Introduction

"A picture is worth one thousand words". This proverb comes from Confucius – a Chinese philosopher before about 2500 years ago. Now, the essence of these words is universally understood. A picture can be magical in its ability to quickly communicate a complex story or a set of ideas that can be recalled by the viewer later in time.

Visual information plays an important role in our society, it will play an increasingly pervasive role in our lives, and there will be a growing need to have these sources processed further. The pictures or images are used in many application areas like architectural and engineering design, fashion, journalism, advertising, entertainment, etc. The emergence of new technologies allows generating information easily under visual forms, leading everyday to an increasing number of generated digital images. Thus it provides the necessary opportunity for us to use the abundance of images. However, the knowledge will be useless if one can't find it. In the face of the substantive and increasing apace images, how to search and to retrieve the images that we interested with facility is a fatal problem: it brings a necessity for image retrieval systems.

Image retrieval consists of the techniques used for query specification and retrieval of images from a digital image collection. It is considered as a part of the field of Information Retrieval (IR), a large and mature research area. IR is the field that deals with "the representation, storage, organization and access to information items". The purpose of information retrieval is to provide the user with easy access to the information items of interest. Image retrieval has become an active research and development domain since the early 1970s [1]. During the last decade the research on image retrieval became of high importance. The most frequent and common means for image retrieval is to index them with text keywords. Text annotation is obtained by manual effort, transcripts, captions, embedded text, or hyperlinked documents. In these systems, keyword and full text searching may be enhanced by natural language processing techniques to provide great potential for categorizing and matching images. Today, some famous text-based image search engines can be found in Internet, for example: Google Image Search[1], Altavista Image Search[2], Corbis[3], Ditto[4], Lycos Image Search[5], Picsearch[6], Behold[7] and Cortina[8], etc.

Text-based search techniques are the most direct, accurate, and effective methods for finding "unconstrained" images. These images are generally organized loosely by category, such as animals, natural scenes, people, and so on. All image indexing is done by human indexers who list the important objects in an image and other terms by which users may wish to access it. Although

[1] http://images.google.com/
[2] http://www.altavista.com/image/
[3] http://pro.corbis.com/
[4] http://www.ditto.com/
[5] http://search.lycos.com/
[6] http://www.picsearch.com/
[7] http://www.behold.cc/
[8] http://cortina.ece.ucsb.edu/

this technique seems to be simple, with the arriving of the era “information explosion", the wide dissemination and use of visual information are inevitable. Every day, thousands upon thousands of visual information, particularly, digital images, come into being. Manual annotation of these images becomes rapidly laborious and fastidious. So it becomes more and more impractical and inefficient while facing to large volumes of images. In addition, images are rich in content, and the capability of manual annotation is limited. For some special domains, for example, crime prevention, medical diagnosis, land analysis, trademark retrieval, traffic sign recognition, etc, very often, textual information is not sufficient. The natural annotating language can not depict an image exactly and minutely.

So to overcome the difficulties due to the huge data volume, people start to pay attention to image itself. As we know, visual features of the images also provide a description of their content. The second image retrieval method, content-based image retrieval (CBIR), emerged as a promising mean for retrieving images and browsing large images databases. Content-based image retrieval has been a topic of intensive research in recent years. It is the process of retrieving images from a collection based on automatically extracted features. It is the part of pattern recognition that can help to resolve many problems such as: optical character recognition (OCR), zip-code recognition, traffic sign recognition, bank check recognition, industrial parts inspection, medical image computer-aided diagnoses, retina recognition, iris recognition, face recognition, fingerprint recognition, palm recognition, document recognition, gait or gesture recognition, image indexing, etc. It allows users search interesting image by specifying image queries. The query can be an example, sketches, visual features (e.g., color, texture and shape), and even the arranging spatial structure of features. One or more features can be used in a specific application. For example, in a satellite information system, the texture features are important, while shape and color features are more important in trademark recognition systems. Once the features have been extracted, the retrieval becomes a task of measuring similarity between image features.

Many content-based image search systems have been developed for various applications. Common query mechanisms provided by user interface are: query by keyword, query by sketching, query by example, browsing by categories, feature selection and weighting, retrieval refining and relevance feedback. Today, some CBIR engines can be found in Internet, for example: Webseek[9], IKONA[10], VIPER[11], CBIR Project[12], ALIPR[13], Tiltomo[14], CIRES[15], RETIN[16], MFIRS[17], Retrievr[18], Multimedia Analysis and Retrieval Systems[19] and Xcavator[20], etc.

Although the image retrieval results by CBIR engines are inferior to the results by text-based engines very much, it is a great realization of the computer vision and machine intelligence. By exploring the synergy between textual and visual features, these image search systems may be further improved.

The generic system architecture for a content-based image retrieval system is included in Figure 1.

[9] http://persia.ee.columbia.edu:8008/
[10] http://www-rocq.inria.fr/cgi-bin/imedia/cbir-gen.cgi/
[11] http://viper.unige.ch/
[12] http://infolab.stanford.edu/IMAGE/
[13] http://www.alipr.com/
[14] http://www.tiltomo.com/
[15] http://amazon.ece.utexas.edu/~qasim/research.htm/
[16] http://www-etis.ensea.fr/Equipes/MIDI/Image/retin-interactive-image-search-engine/
[17] http://www.pilevar.com/mfirs/
[18] http://labs.systemone.at/retrievr/
[19] http://www.ifp.uiuc.edu/~qitian/MARS.html
[20] http://www.xcavator.net/

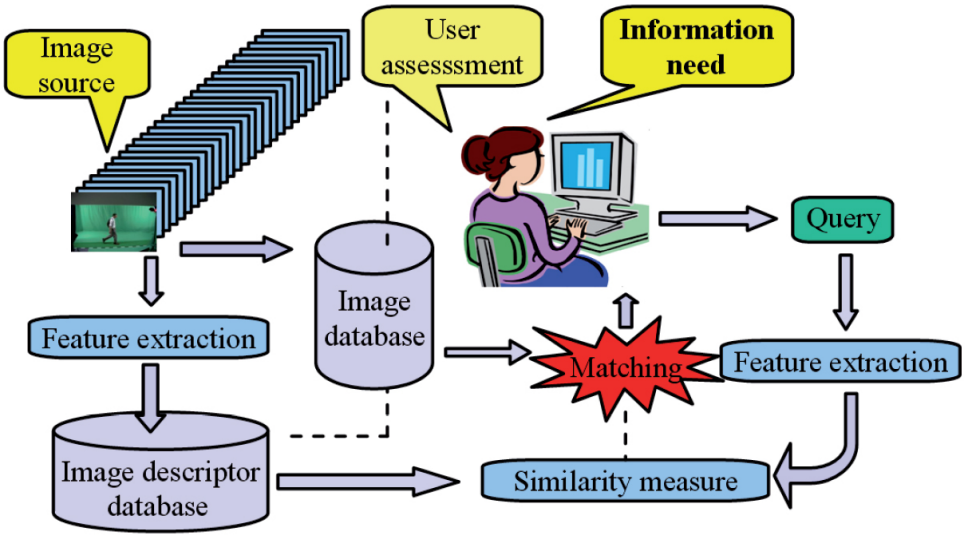


Figure 1: Content-based image retrieval system

Although the closed interaction loop (including users), the supporting database components for retrieval and indexing, the integration with multimedia features, the efficient user interfaces for query specification and image browsing are absolutely necessary in image retrieval system, the essence of a good CBIR system is the methods of feature extraction and similarity measure.

Features extraction is a process of transforming the input data into the set of features. An image includes a large amount of data and it is suspected to be notoriously redundant (much data, but not much information) then the input data of the image will be transformed into a reduced representation set of features (also named features vector). If the extracted features are carefully chosen it is expected that the features set will provide the relevant information from the input data in order to perform the desired task using this reduced representation instead of the full size input. There are two general methods for image classification and retrieval: intensity-based (color and texture) and geometry-based (shape). The representation of the extracted features is the so called descriptor.

Similarity measure is a distance function that associates a numeric value with (a pair of) descriptors.

Among the various forthcoming retrieval systems, we can encounter MPEG-7. Formally named "Multimedia Content Description Interface", MPEG-7 aims at managing data in the way that content information can be retrieved easily. It is under development by the Moving Picture Coding Experts Group (MPEG) that is a working group of ISO/IEC[21] standards organization. It is in charge of the development of international standards for video and/or audio compression, decompression, processing and representation. This group has also developed well-known standards that are MPEG-1, MPEG-2 and MPEG-4. These 3 standards make content available but MPEG-7 enables to find the desired content in a database. MPEG-7 visual description tools consist of basic structure and descriptions that cover basic visual features: color, texture, shape, motion and localization[2, 3]. The use of shape as a visual cue in image retrieval is less developed

[21]Stands for International Standards Organization/International Electro-technical Committee.

than the other visual features, mainly because of the inherent complexity of representing it. Yet, with respect to other features, like color and texture, shape is probably the most important property that is perceived about objects; it is much more effective in semantically characterizing the content of an image. And it is also an important characteristic to identify and distinguish objects [4, 5]. Shape and silhouette provide significant cues for human visual perception. Considerable amount of information exists in two dimensional shapes of objects which enable us to recognize objects without using further information. This has been documented in a vast literature in cognitive psychology or biological neuroscience. It is thus no surprise that shape-based approach is popular in computer vision, too [4, 6]. So retrieval by shape has the potential of being the most effective search technique in many application fields.

There are two main steps involved in shape feature extraction: shape segmentation and shape feature extraction. In order to use shape as a tool for image indexing and retrieval, shapes will often be determined first applying segmentation to an image. Shapes must be identified in, and extracted from an image. Shape segmentation is considered to be one of difficult aspects of image retrieval based on syntactical patterns owing to the complexity of the individual object shape, background and the existence of noise and occultation. Fortunately, there exists a large part of literature describing different techniques and approaches for shape segmentation, such as region growing, edge detection and texture-based techniques. Image segmentation is one of the most important and widely researched topics in image analysis. Accurate segmentation algorithms are fundamental for shape feature extraction. Although this classic topic has been studied in a great deal of literatures, in some cases accurate shape detection will require human intervention because segmentations are very difficult to be completely automate, especially for some miscellaneous images. Image segmentation is beyond the scope of this thesis.

The thesis focuses on shape-based feature extraction and its similarity measure. These technologies have wide applications. Besides image retrieval, they can be developed for object recognition, image alignment and registration, computer vision, image understanding, processing of pictorial information, video/image compression, and so on. However, these technologies are also a difficult problem and the performance of most existing approaches cannot meet people's requirements, especially when applied on a general database. The principal reason is that no current algorithm of shape-based feature extraction and similarity measure is exactly consistent with the perception of human.

We assume that the shapes have been segmented from an image. The representation of a shape may be a boundary contour by a set of coordinates, or be a region (silhouette) by a 2-D function. The goal of shape feature extraction is to attain a shape descriptor.

This descriptor is some set of numbers that are produced to describe a given shape. The shape may not be entirely reconstructed from the descriptor, but the descriptor for different shapes should be different enough that the shapes can be discriminated. The better descriptor is, using certain similarity measure, the greater the difference in the descriptors of significantly different shapes and the lesser the difference for similar shapes. Further more, the better shape descriptor is invariant to shape translation, rotation, scaling, affine and non-rigid transforms; at the same time, it is robust to partial occultation, distortions and corruptions due to the noise. In addition, compactness and simplicity of shape descriptors are necessary for minimizing the storage overhead and the extraction time. Well, we know that it is a notorious difficult task!

Many shape description and similarity measure techniques can be found in the literatures. These techniques adopt different approaches for describing shapes. For general shape description, the reviews are given in [7, 8, 4, 9].

The contents of the thesis are summarized and structured as two parts.

Part 1:

- Study of the fundamental concepts about shapes (Chapter 1).
- Classification and study of the methods of shape-based feature extraction and similarity measure (Chapter 2).

Part 2:

- Proposal of two new affine invariant contour-based descriptors (Chapter 3).
- Proposal of a new shape-based descriptor (Chapter 4).

Then, after these developments, a conclusion will be given.

Part I

Related works

Chapter 1

Fundamental Concepts

In this chapter, the fundamental concepts about shape feature extraction and similarity measure are presented. These concepts involve the shape representations and normalization, shape parameters, the definition of similarity distance of shape descriptors, the techniques of similarity measures and the evaluations of retrieval results.

1.1 Shape representation

A shape of a physical object is its external form or contour, the geometry of its external surfaces or contours, the boundary between the objects interior and the exterior. Shape is the outline or characteristic surface configuration of the object. A shape can be represented by its contour, region or finite points set:

- Contour is only the boundary information that is exploited for shape representation; it is a continuous curve in the plane.
- Region is all the pixels within a shape which are taken into account to obtain the shape representation; it may consist of either a single region or a set of regions as well as some holes in the object.
- Finite point set is the distinct points on the edge of an object which are taken to obtain the shape representation; it holds some important points or lines on a shape that are sensitive to human vision.

So there are 3 research objects for a shape: contour-based, region-based and finite point set-based shape representation [10].

Contour-based shape descriptors make use of only the boundary information, ignoring the shape interior content. Examples to contour-based shape descriptors include Fourier descriptors [11, 12, 13], wavelet descriptors [14, 15, 16], curvature scale space descriptor [17], etc. Since they are computed using only boundary pixels, in general, their computational complexity is low and the sizes of their features are compact. Moreover the contour can be considered as a digital curve, so one can apply a great deal of the computational geometry methods into the analysis of a contour. However, these descriptors cannot well represent shapes for which the complete boundary information is not sufficient or not available such as objects with holes, partially occulted objects, and complex objects consisting of multiple disconnected regions. They may also be easily being corrupted by noise. In addition, as the contour is a set of ordering points

along the boundary of a shape, it is a hard work to segment and extract a contour in a real image.

Region-based shape descriptors exploit both boundary and interior pixels of the shape. Examples of region-based shape descriptors include region moment descriptor [10, 18], grid descriptor [19], shape matrix [20, 21], etc. Since the region-based shape descriptor makes use of all pixels constituting the shape within a frame, it can describe diverse shapes efficiently in a single descriptor. It is also robust to minor deformation along the boundary of the object and to shape distortions. Moreover the segmentation of regions is much easier than that of contours. However, the size of region-based feature is usually great. It leads to an increasing computational complexity of feature extraction and similarity measure.

Finite point set-based shape descriptor represents a set of points sampled from an output of edge detector applied to an object. Finite point set is also called collection of points. Examples of finite point set-based shape descriptors include shape contexts [22], voting schemes [7], etc. They are not required to order the points one by one. They can extract the features of an object with several holes and even with several parts. The key idea of the similarity measure is actually to find the right correspondence of the points between the two shapes. It is very easy to be achieved because of segmenting a set of points of a shape without ordering them. So the finite point sets-based shape descriptor is feasible in practice.

(a)

(b) (c) (d) (e)

(a) Original image; (b) region of object; (c) multi-contours of object; (d) outline contour of object; (e) collection of points of object.

Figure 1.1: Illustration of region, multi-contours, contour and collection of points of an object

Figure 1.1(a) is Olympic Flame, one of 2008 Beijing Olympic mascot. Via region segmentation and edge following, we obtain its region in (b), all of its contours in (c), its outline contour in (d) and one of its collection of points in (e).

Region is a binary image, where each object point is one and the background is zero. In shape recognition field, it is of particular interest to consider the case where the general function $f(x, y)$ is

$$f(x,y) = \begin{cases} 1 & if\ (x,y) \in D \\ 0 & otherwise \end{cases} \tag{1.1}$$

where D is the domain of the binary shape.

Contour is the collection of all the edge points of a region. An object represented by Figure 1.1(c) is called multi-contours representation, and by Figure 1.1(d) is called outline contour representation. Considering the complexity of representation and computation, in general, the contour of an object is its outline contour. In this thesis, all the concepts and methods about contour are the outline contour except specially indicated.

A contour is a closed curve, the discrete parametric equation in Cartesian coordinate system is

$$\Gamma(n) = (x(n), y(n)) \tag{1.2}$$

where $n \in [0, N-1]$; a contour may be parametrized with any number N of vertices and $\Gamma(N) = \Gamma(0)$. For simplicity, we may use $P_n(x_n, y_n)$ instead of $P_n(x(n), y(n))$ to represent a point.

Equal arc length normalization, is also called equidistance normalization. All the distances between a pair of successive vertices on a contour are equal. In the equal arc length parametric equation, $x(s)$ and $y(s)$ are the parametric coordinate functions, sampled at equidistance of values of s. The equal arc length normalization is a simplest and commonest way to represent a contour and it best preserves the boundary topological structure.

The arc length parameter transforms linearly under any linear transformation up to the similarity transform. Translation and rotation do not affect the arc length; scaling scales the parameter by the same amount. The arbitrary choice of a starting point only introduces a shift in the parameter. Due to these invariance properties of the arc length, it is usually used for parametrizing object contours undergoing the similarity transformation. To normalize a contour with equal arc length is not a difficult task; however, there are some remarkable problems. We will talk about them as followings.

We may normalize a contour using linear interpolation to equal arc length parametric equation with the number of vertices equalling M by the following steps:

1. Calculate the arc length function $L(\mu)$ according to original parametric equation Eq. 1.2:

$$L(\mu) = \sum_{n=0}^{\mu} \sqrt{(x_{n+1} - x_n)^2 + (y_{n+1} - y_n)^2} \tag{1.3}$$

where $\mu \in [0, N-1]$ and the last point $P_N(x_N, y_N)$ is assumed to be the same as the first point $P_0(x_0, y_0)$.

So the perimeter $\mathcal{O}$ of the contour is $L(N-1)$.

2. Assume d is the distance of a pair of successive sampled points on the contour with equal arc length normalization, then $d = \mathcal{O}/M$, M is the number of points on normalized contour.

3. Without loss of generality, the first point P'_0 on the normalized contour is the first point P_0 on the original contour. For seeking for the q^{th} ($q \in [1, M-1]$) normalized point P'_q on the contour, the distance between the first point and the point P'_q should be equal to $q{\cdot}d$. So we search for arc length function $L(\mu)$ from $\mu{=}0$, to find the first point whose arc length is greater than $q{\cdot}d$. We assume this point is r^{th} point P_r on the original contour point sequence. The point $P'_q(x'_q, y'_q)$ must be between the $P_r(x_r, y_r)$ and $P_{r-1}(x_{r-1}, y_{r-1})$. Assume a is the distance between the points P_r and P_{r-1}, $a = L(r) - L(r-1)$; b is the distance between the points P_{r-1} and P'_q: $b = q \cdot d - L(r-1)$. Then:

$$\begin{cases} x'_q = \frac{b}{a}(x_r - x_{r-1}) + x_{r-1} \\ y'_q = \frac{b}{a}(y_r - y_{r-1}) + y_{r-1} \end{cases} \tag{1.4}$$

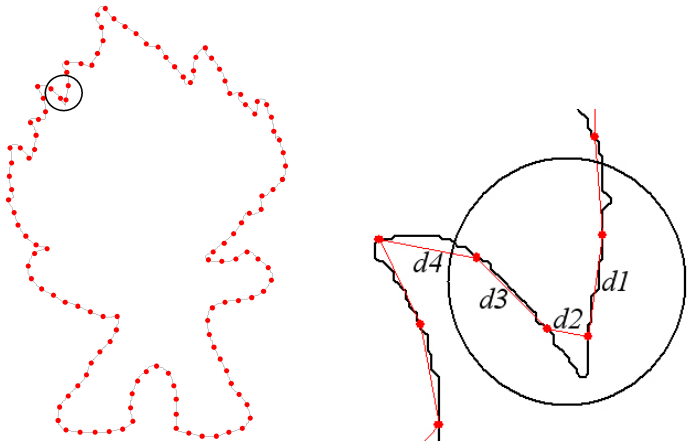
So points collection $P'_s(x'_s, y'_s)$, $s \in [0, M-1]$, composes equidistance normalized contour. The equal arc length contour parametric equation is represented by:

$$\Gamma(s) = (x_s, y_s) \tag{1.5}$$

where $s \in [0, M-1]$.

The difference of a perfect equal arc length contour parametric equation (Eq. 1.5) from general contour parametric equation (Eq. 1.2), consists in that the equal values of s are the equal distance between two successive points.

Analyzing the procedure of the equidistance normalization, we find that the distances of a pair of successive vertices on an equidistance normalized contour by the above method is sometimes not exactly equal, practically, when the contour is badly perturbed by noise. We can find it in Figure 1.2.



(a) (b)

(a) Equidistance normalization by the number 128 of points; the dots are the sampled points by equidistance normalization; (b) zoomed part near the circle in (a): the bold line is the partial original contour; the dots are the sampled points by equidistance normalization.

Figure 1.2: Equidistance normalization

Clearly, the distance $d1, d2, d3$ and $d4$ are not equal in Figure 1.2(b). Let us calculate the standard deviation of the total 128 distances along the contour: assume the mean distance of

two successive vertices is D:

$$D = \frac{1}{N}\sum_{n=0}^{N-1} d_n \tag{1.6}$$

where d_n is the distance between points P_n and P_{n+1}. Then the standard deviation S is:

$$S = \sqrt{\frac{1}{N}\sum_{n=0}^{N-1}(d_n - D)^2} \tag{1.7}$$

In Figure 1.2, D=25.69, S=2.71; then the ratio of standard deviation over mean distance $\frac{S}{D} = 10.5\%$.

The result shows that the ratio of standard deviation over mean distance is over 10% after equidistance normalization. Clearly, using the parameter s replacing the arc length will bring great error when we calculate some contour function, e.g. the curvature function.

Due to the fact that digital contour represented by new sampled points is not the original digital contour, then the line between a pair of new successive vertices is only a chord of the original contour. In general, the length between two points along the original contour is not equal to the length of the chord between the two same points. Theoretically, there are no methods to normalize a digital contour to exact equidistant vertices except the number of normalized points is infinite. What we can do is to try decreasing these errors. The following two methods can improve the normalization precision.

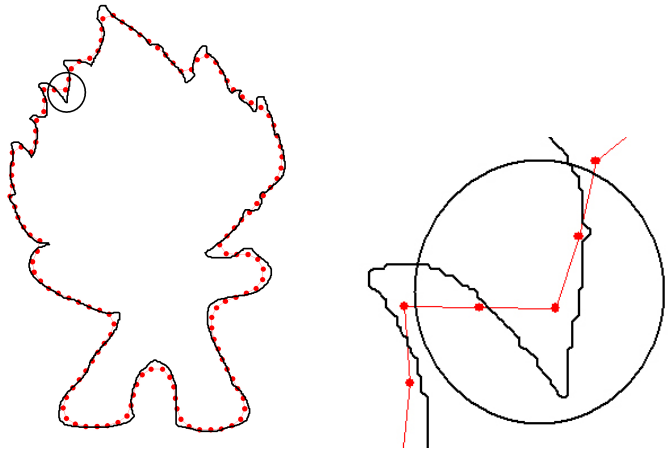
The first method is to repeat the same normalization procedure several times. Table 1.1 gives the results of mean distance, standard deviation and the ratio of standard deviation over mean distance by different number of repeating equidistance normalization.

Number of iterated equidistance normalization	Mean distance D	Standard deviation S	S/D
1	25.69	2.71	10.5%
2	24.51	2.16	8.8%
3	23.75	1.59	6.7%
4	23.21	1.20	5.2%
5	22.65	1.15	5.1%
6	22.19	0.96	4.3%
7	21.94	0.49	2.2%
8	21.84	0.24	1.1%
9	21.79	0.12	0.6%
10	21.74	0.08	0.4%

Table 1.1: Evaluation of S/D with the numbers of iterated normalization

Table 1.1 shows that, increasing the number of iterating equidistance normalization, the distance standard deviation and the ratio of mean deviation distance decrease, i.e. the distances between any two successive points are more and more uniform.

Figure 1.3 is the result of iterating 10 times equidistance normalization. As we see, the distance between a pair of successive vertices is much more uniform than that in Figure 1.2. The cost is that it loses more detail information than a single equidistance normalization in Figure 1.2: this is illustrated by some new normalized points deviating original contour.



(a) (b)

(a) The bold line is the original contour; the dots are the sampled points by iterating 10 times equidistance normalization; (b) zoomed part near the circle in (a).

Figure 1.3: Iterating 10 times equidistance normalization

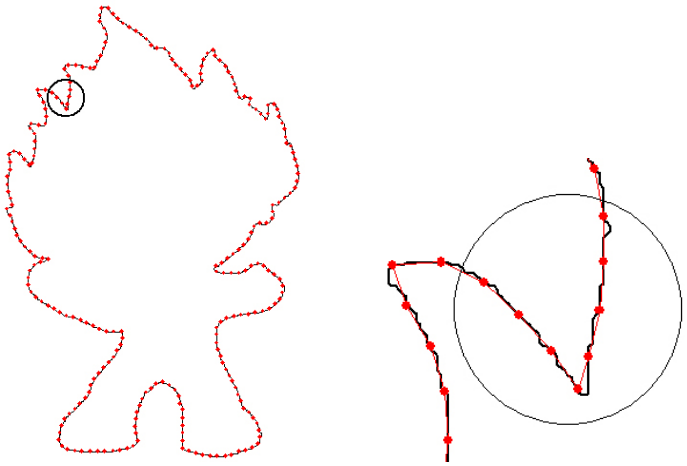
The second method is to increase the number of normalized points. Table 1.2 gives the results of mean distance, standard deviation and ratio of standard deviation over mean distance for equidistance normalization with different number of points. This table shows that, increasing the number of normalized points, decrease the ratio of standard deviation over mean distance S/D, i.e. the distance is more uniform.

The number of normalized points	Mean distance D	Standard deviation D	S/D
256	13.23	0.97	7.3%
512	6.74	0.40	6.0%
768	4.53	0.25	5.5%
1024	3.43	0.16	4.8%
2048	1.75	0.07	3.9%

Table 1.2: Evaluation of S/D with different number of points

Figure 1.4 shows the points set of the contour with 256 normalized points. Comparing with Figure 1.2, the distance is more uniform by increasing N. It preserves almost all the details of its original contour. The cost is also clear: it uses more data to represent a contour. So it will be more computation time consumming.

Therefore, equidistance normalization is not reliable to some contour function related to arc length. To get the precise function values, we will talk about it in Subsection 2.2.4. Although arc length parameter is invariant to shape translation, rotation and scaling, it is nonlinearly transformed under an affine transformation, and would not be a suitable parameter in this situation. We will propose a new normalization for affine invariance in Chapter 3.



(a) (b)

(a) The dots are the sampled points by equidistance normalization; (b) zoomed part near the circle in (a).

Figure 1.4: Equidistance normalization to 256 points

1.2 Shape parameters

Basically, shape-based image retrieval consists of the measuring of similarity between shapes represented by their features. Some simple geometric features can be used to describe shapes. Usually, the simple geometric features can only discriminate shapes with large differences; therefore, they are usually used as filters to eliminate false hits or combined with other shape descriptors to discriminate shapes. They are not suitable to be stand alone shape descriptors. A shape can be described by different aspects. These shape parameters are Center of gravity, Axis of least inertia, Digital bending energy, Eccentricity, Circularity ratio, Elliptic variance, Rectangularity, Convexity, Solidity, Euler number, Profiles, Hole area ratio. They will be introduced in this section.

1.2.1 Center of gravity

The center of gravity is also called centroid. Its position should be fixed in relation to the shape.

If a shape is represented by its region function Eq. 1.1, its centroid (g_x, g_y) is:

$$\begin{cases} g_x = & \frac{1}{N}\sum_{i=1}^{N} x_i \\ g_y = & \frac{1}{N}\sum_{i=1}^{N} y_i \end{cases} \tag{1.8}$$

where N is the number of point in the shape, $(x_i, y_i) \in \{(x_i, y_i) \mid f(x_i, y_i) = 1\}$.

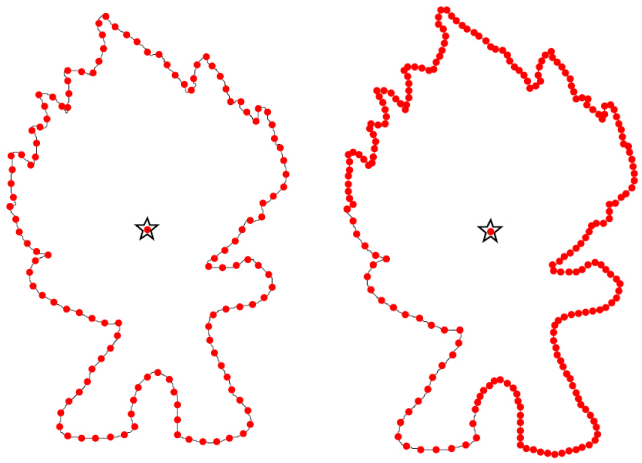
If a shape is represented by its contour Eq. 1.2, the position of its centroid is given below:

$$\begin{cases} g_x = \frac{1}{6A}\sum_{i=0}^{N-1}(x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \\ g_y = \frac{1}{6A}\sum_{i=0}^{N-1}(y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \end{cases} \tag{1.9}$$

where A is the contour's area and

$$A = \frac{1}{2}\left|\sum_{i=0}^{N-1}(x_i y_{i+1} - x_{i+1} y_i)\right| \tag{1.10}$$

The position of shape centroid is fixed with different points distribution on a contour. One can notice that the position of the centroid in Figure 1.5 is fixed no matter how the points distribution is.



(a) (b)

The dots are points distributed on contour by uniformly (a) and non-uniformly (b). The star is the centroid of original contour; the inner dot is the centroid of sampled contour.

Figure 1.5: Centroid of contour

So using Eq. 1.9, we can obtain the genuine centroid of a contour under whatever the contour is normalized.

1.2.2 Axis of least inertia

The axis of least inertia is unique to the shape. It serves as a unique reference line to preserve the orientation of the shape. The axis of least inertia (ALI) of a shape is defined as the line for which the integral of the square of the distances to points on the shape boundary is a minimum.

Since the axis of inertia pass through the centroid of a contour, to find the ALI, transfer the shape and let the centroid of the shape be the origin of Cartesian coordinate system. Let $x\sin\theta - y\cos\theta = 0$ be the parametric equation of ALI. The slope angle θ is estimated as follows:

Let α be the angle between the axis of least inertia and the x-axis. The inertia is given by [23, 24]:

$$I = \frac{1}{2}(a+c) - \frac{1}{2}(a-c)\cos(2\alpha) - \frac{1}{2}bsin(2\alpha)$$

where $a = \sum_{i=0}^{N-1} x_i^2$, $b = 2\sum_{i=0}^{N-1} x_i y_i$, $c = \sum_{i=0}^{N-1} y_i^2$.

Hence,

$$\frac{dI}{d\alpha} = (a-c)\sin(2\alpha) - b\cos(2\alpha)$$

$$\frac{d^2I}{d\alpha^2} = 2(a-c)\cos(2\alpha) + 2b\sin(2\alpha)$$

Let $dI/d\alpha = 0$, we obtain

$$\alpha = \frac{1}{2}\arctan\left(\frac{b}{a-c}\right),\ -\frac{\pi}{2} < \alpha < \frac{\pi}{2}$$

The slope angle θ is given by

$$\theta = \begin{cases} \alpha + \frac{\pi}{2} & if\ \frac{d^2I}{d\alpha^2} < 0 \\ \alpha & otherwise \end{cases}$$

1.2.3 Average bending energy

Average bending energy BE is defined by

$$BE = \frac{1}{N} \sum_{s=0}^{N-1} K(s)^2$$

where *K(s)* is the curvature function, s is the arc length parameter, and N is the number of points on a contour [4]. In order to compute the average bending energy more efficiently, Young et. al. [25] did the Fourier transform of the boundary and used Fourier coefficients and Parseval's relation.

One can prove that the circle is the shape having the minimum average bending energy.

1.2.4 Eccentricity

Eccentricity is the measure of aspect ratio. It is the ratio of the length of major axis to the length of minor axis. It can be calculated by principal axes method or minimum bounding rectangle method.

Principal axes method

Principal axes of a given shape can be uniquely defined as the two segments of lines that cross each other orthogonally in the centroid of the shape and represent the directions with zero cross-correlation [26]. This way, a contour is seen as an instance from a statistical distribution. Let us consider the covariance matrix C of a contour:

$$C = \frac{1}{N} \sum_{i=0}^{N-1} \begin{pmatrix} x_i - g_x \\ y_i - g_y \end{pmatrix} \begin{pmatrix} x_i - g_x \\ y_i - g_y \end{pmatrix}^T = \begin{pmatrix} c_{xx} & c_{xy} \\ c_{yx} & c_{yy} \end{pmatrix} \tag{1.11}$$

where

$c_{xx} = \frac{1}{N} \sum_{i=0}^{N-1} (x_i - g_x)^2$
$c_{xy} = \frac{1}{N} \sum_{i=0}^{N-1} (x_i - g_x)(y_i - g_y)$
$c_{yx} = \frac{1}{N} \sum_{i=0}^{N-1} (y_i - g_y)(x_i - g_x)$
$c_{yy} = \frac{1}{N} \sum_{i=0}^{N-1} (y_i - g_y)^2$

$G(g_x, g_y)$ is the centroid of the shape. Clearly, here $c_{xy} = c_{yx}$.

The lengths of the two principal axes equal the eigenvalues λ_1 and λ_2 of the covariance matrix C of a contour, respectively.

So the eigenvalues λ_1 and λ_2 can be calculated by

$$det(C - \lambda_{1,2} I) = det \begin{pmatrix} c_{xx} - \lambda_{1,2} & c_{xy} \\ c_{yx} & c_{yy} - \lambda_{1,2} \end{pmatrix} = (c_{xx} - \lambda_{1,2})(c_{yy} - \lambda_{1,2}) - c_{xy}^2 = 0$$

So

$$\begin{cases} \lambda_1 = \frac{1}{2}\left[c_{xx} + c_{yy} - \sqrt{\left(c_{xx} + c_{yy}\right)^2 - 4\left(c_{xx}c_{yy} - c_{xy}^2\right)}\right] \\ \lambda_2 = \frac{1}{2}\left[c_{xx} + c_{yy} + \sqrt{\left(c_{xx} + c_{yy}\right)^2 - 4\left(c_{xx}c_{yy} - c_{xy}^2\right)}\right] \end{cases}$$

Then, eccentricity can be calculated:

$$E = \lambda_2/\lambda_1 \tag{1.12}$$

Minimum bounding rectangle

Minimum bounding rectangle is also called minimum bounding box. It is the smallest rectangle that contains every point in the shape. For an arbitrary shape, eccentricity is the ratio of the length L and width W of minimal bounding rectangle of the shape at some set of orientations. Elongation, *Elo*, is an other concept based on eccentricity (cf. Figure 1.6):

$$Elo = 1 - W/L \tag{1.13}$$

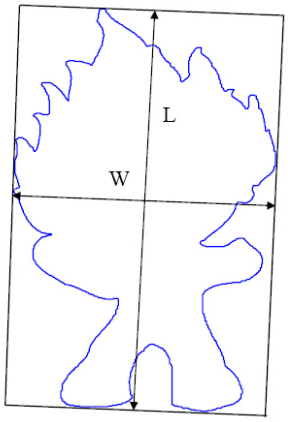


Figure 1.6: Minimum bounding rectangle and corresponding parameters for elongation

Elongation is a measure that takes values in the range $[0, 1]$. A symmetrical shape in all axes such as a circle or square will have an elongation value of 0 whereas shapes with large aspect ratios will have an elongation closer to 1.

1.2.5 Circularity ratio

Circularity ratio represents how a shape is similar to a circle [8]. There are 3 definitions:

- Circularity ratio is the ratio of the area of a shape to the area of a circle having the same perimeter:

$$C_1 = \frac{A_s}{A_c} \tag{1.14}$$

  where A_s is the area of the shape and A_c is the area of the circle having the same perimeter as the shape. Assume the perimeter is $\mathcal{O}$, so $A_c = \mathcal{O}^2/4\pi$. Then $C_1 = 4\pi \cdot A_s/\mathcal{O}^2$. As 4π is a constant, so we have the second circularity ratio definition.

- Circularity ratio is the ratio of the area of a shape to the shape's perimeter square:

$$C_2 = \frac{A_s}{\mathcal{O}^2} \tag{1.15}$$

- Circularity ratio is also called circle variance, and defined as:

$$C_{va} = \frac{\sigma_R}{\mu_R} \tag{1.16}$$

where μ_R and σ_R are the mean and standard deviation of the radial distance from the centroid (g_x, g_y) of the shape to the boundary points $(x_i, y_i), i \in [0, N-1]$. They are the following formulae respectively:

$$\mu_R = \frac{1}{N}\sum_{i=1}^{N-1} d_i \quad and \quad \sigma_R = \sqrt{\frac{1}{N}\sum_{i=1}^{N-1}(d_i - \mu_R)^2}$$

where $d_i = \sqrt{(x_i - g_x)^2 + (y_i - g_y)^2}$.

The most compact shape is a circle. See Figure 1.7.

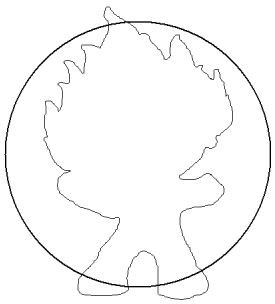


Figure 1.7: Circle variance

1.2.6 Ellipse variance

Ellipse variance E_{va} is a mapping error of a shape to fit an ellipse that has an equal covariance matrix as the shape: $C_{ellipse} = C$ (cf. Eq.1.11). It is practically effective to apply the inverse approach yielding.

We assume

$$V_i = \begin{pmatrix} x_i - g_x \\ y_i - g_y \end{pmatrix}$$

$$d'_i = \sqrt{V_i^T \cdot C_{ellipse}^{-1} \cdot V_i}$$

$$\mu'_R = \frac{1}{N}\sum_{i=1}^{N-1} d'_i \quad and \quad \sigma'_R = \sqrt{\frac{1}{N}\sum_{i=1}^{N-1}(d'_i - \mu'_R)^2}$$

Then

$$E_{va} = \frac{\sigma'_R}{\mu'_R} \tag{1.17}$$

Comparing with Eq. 1.16, intuitively, E_{va} represents a shape more accurately than C_{va}, cf. Figure 1.8.

Figure 1.8: Ellipse variance

1.2.7 Rectangularity

Rectangularity represents how rectangular a shape is, i.e. how much it fills its minimum bounding rectangle:

$$Rectangularity = A_S/A_R$$

where A_S is the area of a shape; A_R is the area of the minimum bounding rectangle.

1.2.8 Convexity

Convexity is defined as the ratio of perimeters of the convex hull $\mathcal{O}_{Convexhull}$ over that of the original contour $\mathcal{O}$ [26]:

$$Convexity = \frac{\mathcal{O}_{Convexhull}}{\mathcal{O}} \tag{1.18}$$

Figure 1.9: Illustration of convex hull

The region R^2 is a convex if and only if for any two points $P_1, P_2 \in R^2$, the whole line segment P_1P_2 is inside the region. The convex hull of a region is the smallest convex region including it. In Figure 1.9, the outline is the convex hull of the region.

In [26], the authors presented the algorithm for constructing a convex hull by traversing the contour and minimizing turn angle in each step.

1.2.9 Solidity

Solidity describes the extent to which the shape is convex or concave [27] and it is defined by

$$Solidity = A_s/H$$

where, A_s is the area of the shape region and H is the convex hull area of the shape. The solidity of a convex shape is always 1.

1.2.10 Euler number

Euler number describes the relation between the number of contiguous parts and the number of holes on a shape. Let S be the number of contiguous parts and N be the number of holes on a shape. Then the Euler number is:

$$Eul = S - N$$

For example

Euler Number equal to *1, -1* and *0*, respectively.

1.2.11 Profiles

The profiles are the projection of the shape to x-axis and y-axis on Cartesian coordinate system. We obtain two one-dimension functions:

$$Pro_x(i) = \sum_{j=j_{min}}^{j_{max}} f(i,j) \quad and \quad Pro_y(j) = \sum_{i=i_{min}}^{i_{max}} f(i,j)$$

where $f(i, j)$ represents the region of shape Eq. 1.1. See Figure 1.10.

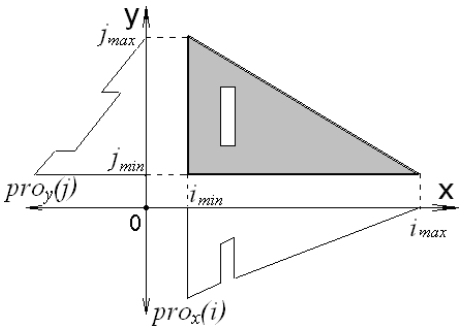


Figure 1.10: Profiles

1.2.12 Hole area ratio

Hole area ratio HAR is defined as

$$HAR = \frac{A_h}{A_s}$$

where A_s is the area of a shape and A_h is the total area of all holes in the shape. Hole area ratio is most effective in discriminating between symbols that have big holes and symbols with small holes [28].

The properties about translation, scaling and rotation of these simple geometric features are summarized in Table 1.3.

Simple Geometric Features	Invariance		
	Translation	Scaling	Rotation
Average bending energy	√	×	√
Eccentricity	√	√	√
Circularity ratio	√	√	√
Ellipse variance	√	√	√
Rectangularity	√	√	√
Convexity	√	√	√
Solidity	√	√	√
Euler Number	√	√	√
Profiles	√	√	×
Hole area ratio	√	√	√

Table 1.3: Properties of simple geometric features

1.3 Shape similarity measures

Shape matching is a central problem in visual information systems, computer vision, pattern recognition and robotics. Shape matching deals with transforming a pattern, and measuring the resemblance with another pattern using some metrics. Shape similarity measures are an essential ingredient in shape matching. We can observe two particular research directions in this area, however, both are essential. The first one follows the subject of retrieval effectiveness, where the goal is to achieve query results complying with the user's expectations (measured by the precision and recall scores). As the effectiveness is obviously dependent on the semantics of similarity measure, we require the possibilities of similarity measuring as rich as possible, thus, the measure should not be limited by properties regarded as restrictive for similarity modeling. Following the second direction, the similarity measures should be as efficient (or fast) as possible, because the number of objects in a database can be large and the similarity scores are often expensive to compute [29, 30, 31].

Shape similarity measures should be in accord with out visual perception. This property leads to the following requirements:

- it should be universal in the sense that it allows us to identify or distinguish objects of arbitrary shapes, i.e., no restrictions on shapes are assumed.

- it should permit recognition of perceptually similar objects that are not mathematically identical.
- it should abstract from distortions, e.g. digitization noise, segmentation errors and partial occultation.
- it should not depend on scaling, orientation, position of objects, affine transform and perspective distortions.

In mathematics, the similarity and dissimilarity measure are defined as following [31]:

Let a model object $O \in U$, where U is a universal model. Let $s : U \times U \mapsto R$ be a similarity measure, where $s(O_i, O_j)$ is considered as a similarity score of objects O_i and O_j. In many cases it is more suitable to use a dissimilarity measure $d : U \times U \mapsto R$ equivalent to a similarity measure s as $s(Q, O_i) > s(Q, O_j) \Longleftrightarrow d(Q, O_i) < d(Q, O_j)$.

In this section different similarity measurements are described in details.

1.3.1 Distance metric space

It is known that human observers judge images as similar if they show similar objects. The computer vision judges objects similarity by computing the distance of shape features. A similarity measure assigns a lower distance (or higher score) to more similar objects, and vice versa. In general, a metric distance is necessary to effectively organize database items through multidimensional vectors. The distance metric space is defined as following [5]: a space $\mathbb{R}^N$ is a metric space if for any of its two elements x and y, there exists a distance $d(x, y)$, that satisfies the following properties:

- $d(x, y) \geq 0$ (non-negativity or positivity)
- $d(x, y) = 0$ if and only if $x = y$ (identity or reflexivity)
- $d(x, y) = d(y, x)$ (symmetry)
- d is continuous
- one major difference should cause a greater dissimilarity than some minor ones
- $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality)

This triangle inequality is a kind of transitivity property; it says if x, y are similar and y, z are similar, then also x and z are similar.

The representation of shape feature is usually called shape descriptor. A shape descriptor is comprised by a set of vectors. So shape similarity distance is essentially the distance of vectors. Assuming $A = (a_0, a_1, \ldots, a_{n-1})$ and $B = (b_0, b_1, \ldots, b_{n-1})$ are the query and target feature vectors respectively, there are many classical methods to compute the vector distance in metric space.

Minkowski-form distance metric (L_p)

The Minkowski-form distance is defined based on the L_p norm as:

$$d_p(A,B) = \left(\sum_{i=0}^{N-1} |a_i - b_i|^{\,p} \right)^{1/p} \tag{1.19}$$

When $p = 1$, $d_1(A,B)$ is 1-*norm distance* L_1. It more colorfully called *city block distance, taxicab norm* or *Manhattan distance*, because it is the distance a car would drive in a city laid out in square blocks (if there are no one-way streets). *Manhattan distance* d_{Man} between two vectors *A, B* measured along axes at right angles. See Figure 1.11(a).

$$d_{Man}(A,B) = \sum_{i=0}^{N-1} \mid a_i - b_i \mid \tag{1.20}$$

When $p = 2$, $d_2(A,B)$ is *Euclidean distance* d_{Eu}. It is the "ordinary" distance between two vectors. See Figure 1.11(b).

$$d_{Eu}(A,B) = \sqrt{\sum_{i=0}^{N-1} (a_i - b_i)^2} \tag{1.21}$$

When $p \to \infty$, we get *infinity norm distance* L_∞. It is also called *Chebyshev distance*. *Chebyshev distance* is a metric defined on a vector space where the distance between two vectors is the greatest of their differences along any coordinate dimension. It is also known as *chessboard distance*: See Figure 1.11(a).

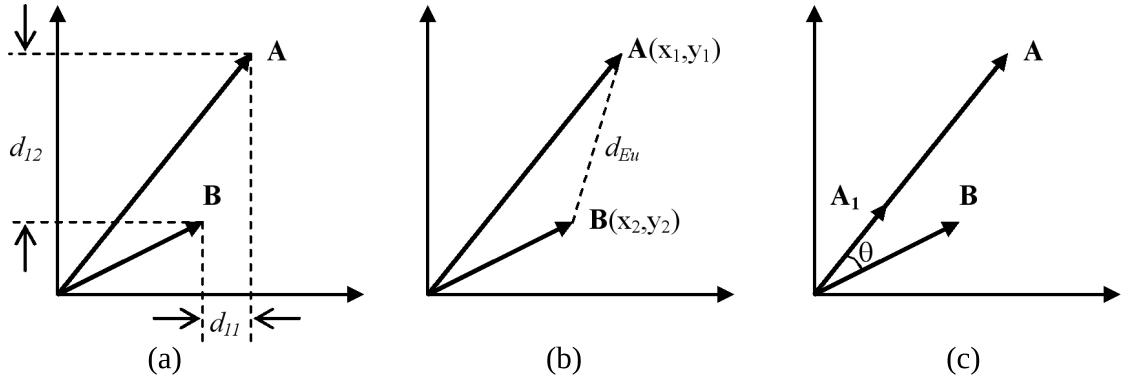
$$L_\infty(A,B) = \max_{0 \le i \le N-1} \{ \mid a_i - b_i \mid \} \tag{1.22}$$

Cosine distance

The *cosine distance* computes the difference in direction, irrespective of vector lengths. The distance is given by the angle between the two vectors.

$$d_{cos}(A,B) = 1 - cos\theta = 1 - \frac{A \cdot B^T}{\|A\| \cdot \|B\|} = 1 - \frac{\sum_{i=0}^{N-1} a_i b_i}{\sqrt{\sum_{i=0}^{N-1} a_i^2} \sqrt{\sum_{i=0}^{N-1} b_i^2}} \tag{1.23}$$

Cosine distance only takes angle into account. As a result, two vectors with the same direction have the same cosine distance to any other vector, see Figure 1.11(c).



(a) Manhattan distance $d_{Man} = d_{11} + d_{12}$ and infinity norm distance $L_\infty = max(d_{12}, d_{11}) = d_{12}$; (b) Euclidean distance $d_{Eu} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$; (c) vector A and A_1 have the same cosine distance to vector B.

Figure 1.11: Distance definition

χ^2 statistics distance

The χ^2 *statistics distance* is defined as

$$D_{\chi^2}(A, B) = \frac{1}{2N} \sum_{i=0}^{N-1} \frac{(a_i - b_i)^2}{(a_i + b_i)}$$

Measures how unlikely it is that one element in a vector is drawn from the population represented by the other.

Kullback-Leibler Divergence distance (KL)

$$D_{KL}(A, B) = \sum_{i=0}^{N-1} a_i log \frac{a_i}{b_i}$$

It is robust with respect to noise.

Jeffrey Divergence distance (JD)

$$D_{JD}(A, B) = \sum_{i=0}^{N-1} \left(a_i log \frac{a_i}{b_i} + b_i log \frac{b_i}{a_i} \right)$$

It is symmetric regarding KL.

Quadratic-form distance (QF)

Considering similarity across elements in vectors, the definition of *quadratic-form distance* is following:

$$D_{QF}(A, B) = \sqrt{(A - B) \cdot S \cdot (A - B)^T}$$

where $S = [s_{ij}]$ is a similarity matrix, s_{ij} denotes similarity (ground distance) between elements a_i and b_j. Some ground distance functions are defined as the following:

(a) $s_{ij} = 1 - d_{ij}/d_{max}$, where d_{ij} is the distance between a_i (i^{th} element in A) and b_j (j^{th} element in B), d_{max} is maximum d_{ij}.

(b) $s_{ij} = exp(-\sigma(d_{ij}/d_{max})^2)$, where σ is positive constant; d_{ij} is same as that in (a).

Mahalanobis Distance

The *Mahalanobis distance* takes into account the covariance among the variables in calculating distances. With this measure, the problems of scale and correlation inherent in the *Euclidean distance* are no longer an issue. To understand how this works, consider that, when using *Euclidean distance*, the set of equidistant points from a given location is a sphere. The *Mahalanobis distance* stretches this sphere to correct for the respective scales of the different variables, and to account for correlation among variables.

The *Mahalanobis distance* between two vectors A and B from the same distribution which has covariance matrix C is defined as:

$$d_{Man} = \sqrt{(A-B)\cdot C^{-1}\cdot (A-B)^T}$$

Hausdorff distance

Informally, the *Hausdorff distance* between two sets of vectors, A and B, is the longest distance you would have take from an arbitrary point in A to the corresponding nearest point in B, i.e. choose a point in A, and the distance to the closest point in B is equal to or less than the Hausdorff distance. See Figure 1.12.

This distance function:

$$d_{Hau}(A,B) = \max\left\{\max_{b\in B}\left[\min_{a\in A}\left(d(a,b)\right)\right], \max_{a\in A}\left[\min_{b\in B}\left(d(a,b)\right)\right]\right\}$$

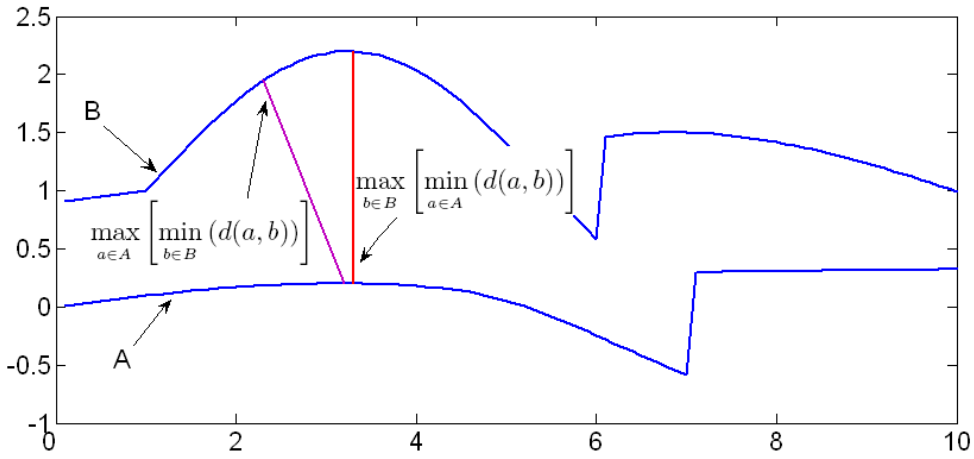


Figure 1.12: Hausdorff distance

The Hausdorff distance is not transformation invariant and also not robust against noise. The advantage of using this metric is that partial matching is possible.

Edit distance

Edit distance between two objects is defined as the number of basic operations (insertion, deletion and substitution) needed to transform one representation into another [32]. It is for computation between a pair of ordered sets of vectors. This way, the differences between shapes can be computed in terms of editing costs.

To measure edit distance, we use the methods of *Hamming distance, Levenshtein distance, Damerau-Levenshtein distance and Jaro-Winkler distance, etc.*

1.3.2 Non-metric distance space

We have the experience: a child resemble somewhat his mother as well as his father. However, in general, the parents do not resemble each other. That is: both the similarity distances between the child and his mother $d(Child, Mother)$ and the similarity distances between the child and his father $d(Child, Father)$ are small; whereas the similarity distances between the parents $d(Mother, Father)$ is large. So we can not assure that it holds the triangle inequality: $d(Child, Mother)+d(Child, Father) \geq d(Mother, Father)$. Let us look at the other example given in Figure 1.13 [33]: as the man and the centaur are partial similar, so the distance from the man to the centaur is small; similarly, the centaur and the horse are partial similar, the distance from the centaur to the horse is also small, but, clearly the distance from the man to the horse is large. In general, the result is: *d(man, centaur)+d(centaur, horse)<d(man, horse).* It does not hold the triangle inequality. These two examples show that human similarity perception does not hold in metric space.

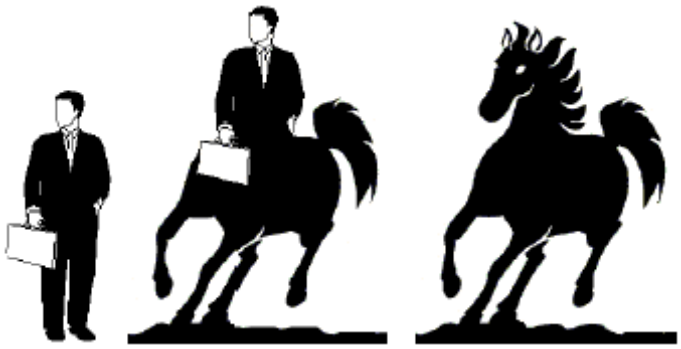


Figure 1.13: Illustration of non-respect of the triangle inequality

As a matter of fact, the distance between two shapes, as perceived by human beings, is not a metric distance. Non-metric measures have been used mainly in the areas of multimedia databases and information retrieval. A common reason for their usage is their better robustness—a robust measure is resistant to outliers, that is, to anomalous or "noisy" objects and it can neglect some portions of the measured objects which appear the most dissimilar.

Elastic matching distance is non-metric distance in common use. For similarlity-based measure, the two shapes are similar with perception of human. In some cases, one of these two shapes may be a transformation of the other shape by rigid-transform which can be represented exactly by mathematic methods, for example, translation, rotation, scaling, shear and even perspective. Whereas in more other cases, this transformation can't be represented mathematically, for example, the non-rigid transform, distortion by noise and\or occultation. To get a better result of similarlity-based measure, the elastic matching is a very efficient method. To realize the elastic

matching, dynamic space warping (DSW) is a very effective algorithm which based on dynamic programming. Unlike the Euclidean distance, DSW is based on non-rigid alignment between the points of the two vectors. Now, we present a brief description of the DSW algorithm.

Let $A = \{\mathbf{a_1}, \mathbf{a_2}, \cdots, \mathbf{a_N}\}$ and $B = \{\mathbf{b_1}, \mathbf{b_2}, \cdots, \mathbf{b_N}\}$ be two finite sets, $\mathbf{a}_\alpha$ and $\mathbf{b}_\alpha$, $\alpha \in [1, N]$, are the scalars or vectors. To align two sequences using DTW we construct an N-by-N matrix D where the (i^{th}, j^{th}) element of the matrix contains the distance $d(\mathbf{a_i}, \mathbf{b_j})$ between the two points $\mathbf{a_i}$ and $\mathbf{b_j}$ (This distance can be defined by any ways; typically the Euclidean distance is used). Each matrix element (i, j) corresponds to the alignment between the points $\mathbf{a_i}$ and $\mathbf{b_j}$. Then, another N-by-N matrix, DT, is constructed to find the optimal correspondence between the points of the two sets. This matrix DT is also called DSW table. The columns of DT represent the points of one set and the rows represent the points of the other. Initially, the elements of DT are set as

$$DT_{initial}(n, m) = \begin{cases} 0, & \max(1, n - w + 1) \leqslant m \leqslant \min(N, n + w - 1) \\ \infty, & otherwise \end{cases}$$

where $n, m \in \langle 1,\ N \rangle$, w is a predefined diagonal width for DT as illustrated in Figure 1.14, and $\max(\alpha,\ \beta)$ and $\min(\alpha,\ \beta)$ are the maximum and minimum values of α and β, respectively. Only the elements of DT that fall within w are updated during the DSW search. This initialization of DT avoids computing the distances between all the points of two sets and restricts the distance computation to only those points which are more likely to correspond to each other. Therefore, the computational complexity is largely reduced while more meaningful correspondences are obtained.

Starting at an arbitrary point for both sets A and B, the DSW table DT is searched through a predefined diagonal window of width w, left-to-right and top-to-bottom, starting from the upper-left element, as shown in Figure 1.14.

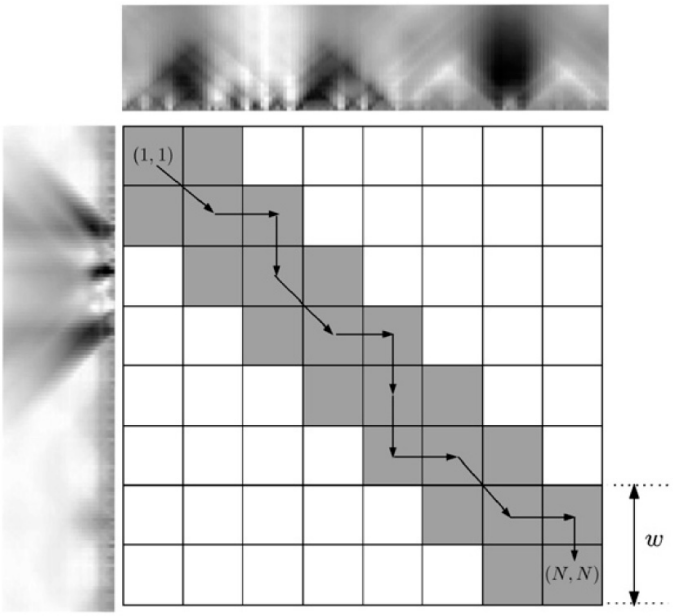


Figure 1.14: Dynamic space warping (DSW) table

The first row and first column elements are initialized as the distance d between the corre-

sponding points. Then, the rest of the w-diagonal elements of DT are updated as

$$DT(n,m) = D(n,m) + \min \begin{cases} DT(n-1,m) \\ DT(n-1,m-1) \\ DT(n,m-1) \end{cases}$$

where n, $m \in \langle 1,\ N \rangle$.

The least cost path through the DSW table DT is the value of element $DT(N,N)$, which corresponds to the best matching between the two sets according to the selected starting points. However, it is clear that the established correspondence is sensitive to the starting point of each set. In order to achieve starting point (or rotation) invariance, it is sufficient to fix the starting point of one set and shift the other set point by point. Moreover, invariance to the mirror transformation can be obtained by flipping the points of one set and repeat the search for the N starting points again. The final least cost correspondence is taken as the minimum value of $DT(N,N)$ among all $2N$ runs of the DSW table search, denoted by DT_{min}. Then, the elastic matching distance

$$d_{DSW} = DT_{min}/L_{AB}$$

where L_{AB} is the length of the least-cost path through the DSW table. In Figure 1.14 for example, the least-cost path passes through 12 grids, so L_{AB}=12.

Generally, we can obtain a non-metric distance by a linear combination of other distances where at least one is non-metric, or we can multiply non-metric distances. Some examples of non-metric distance are represented in [34].

On the other hand, as introduced above, a metric distance is necessary to effectively organize database items through multidimensional index structures. The two different metric contrasting requirements are therefore set for shape matching. They are concerned respectively, with fitting human perception and with the efficiency of data organization and indexing. To cope with both requirements two distinct distance measures have been defined by [5]: using distance metric space to measure partial similarity and non-metric space to measure total similarity.

The desirable distance measure should reflect human perception. That is to say, perceptually similar images should have smaller distance between them and perceptually different images should have larger distance between them. So to choose the method of similarity measure, it depends on the required properties, the particular matching problem and the application.

1.4 Evaluations of retrieval results

To evaluate the performance of different algorithms for image retrieval, an effective performance measurement is necessary. The performance measurements are usually based on statistics of the subjective tests. Different performance measurements often use different subjective tests, resulting in different definitions of retrieval accuracy. In this section we describe some common retrieval effectiveness measurement.

1.4.1 Recall and precision pair

The recall and precision pair (RPP) is perhaps the most widely used retrieval performance measurement in literature. For each query, the system returns a ranked list. Each item in the list is determined as either relevant or not according to the subject test results. Then the effectiveness is measured by recall and precision. Recall measures the ability of retrieving relevant

items from the database. It is defined as the ratio of the number of relevant items retrieved over the total number of relevant items in the database. Precision measures the retrieval accuracy and is defined as the ratio of the number of relevant items retrieved over the number of total items retrieved.

$$recall = \sharp(relevant \quad items \quad retrieved)/\sharp(relevant \quad items \quad in \quad database)$$

$$precision = \sharp(relevant \quad items \quad retrieved)/\sharp(retrieved \quad items)$$

$\sharp\{\alpha\}$ denotes getting the number of the value α.

If we assume a is the number of relevant items retrieved, b is the number of non-relevant items retrieved, c is the number of relevant items non-retrieved in a database, then

$$recall = \frac{a}{a+c} \quad and \quad precision = \frac{a}{a+b}$$

So $recall \in [0,1]$, $precision \in [0,1]$.

The precision normally decreases while increasing the recall. This is because in the process of trying to retrieve all relevant items to a query, some irrelevant items are also retrieved. Thus the recall and the corresponding precision value are used together to indicate the retrieval effectiveness of a system.

1.4.2 Percentage of weighted hits

The percentage of weighted hits (PWH) is similar to the recall measurement in RPP. However, instead of measuring recall based on binary relevance value as in RPP, PWH assigns a weighted relevance value w_i to each item in the dataset [35]. The weight w_i is manually obtained as follows: we have a group of humans and a database. Initially, all items in the database start with zero weight. Each time a human selects a set of items which resemble to a certain query, the weights of these items for this query are increased by one. Thus, the weight w_i of an item is equal to the number of humans who selected it as relevant or similar to a query. In this way, the items that many humans pick have a greater weight than those items which are chosen only little times.

For each query, a fixed number of items are retrieved by a system. Then the percentage of weighted hits P is defined as the following:

$$P = \frac{\sum_{i=1}^{n} w_{\pi(i)}}{\sum_{j=1}^{N} w_j}$$

where n is the number of items retrieved by the system; $\pi(i)$ is the items retrieved by a test system; N is the total number of items in the database; $w_{\pi(i)}$ and w_j are the weights of subjects selecting item $\pi(i)$ and j as relevant to the query.

To indicate the average performance of a system, many queries should be performed and the average percentage of weighted hits over these queries is used as the retrieval effectiveness of the system.

1.4.3 Percentage of similarity ranking

In percentage of similarity ranking (PSR) measurement, each subject assigns a similarity rank to each item in the dataset based on the item's similarity to the query, rather than only assigns relevance/irrelevance as in RPP and PWH [36]. The final result of the subject test is a matrix $\{Q_j(i,k)\}$, where $Q_j(i,k)$ indicates the percentage of people that ranked i^{th} item in the k^{th} position to query j. From $Q_j(i,k)$, we can calculate the mean value $\bar{p_j}(i)$ and standard deviation $\sigma_j(i)$, representing the average ranking of the i^{th} item to given query j and a measure of agreement about a ranking close to $\bar{p_j}(i)$, respectively. If for a query j, a retrieval algorithm returns an item i at rank $P_j(i)$, then the agreement between the algorithm ranking and human ranking is measured by the percentage similarity ranking $S_j(i)$:

$$S_j(i) = \sum_{k=P_j(i)-\frac{\sigma_j(i)}{2}}^{P_j(i)+\frac{\sigma_j(i)}{2}} Q_j(i,k)$$

The plot of $S_j(i)$ as a function of $P_j(i)$ shows the retrieval performance of a retrieval algorithm. High $S_j(i)$ indicates high retrieval accuracy of the algorithm.

1.4.4 Average of normalized modified retrieval rank

The average of normalized modified retrieval rank (ANMRR) is an objective measure used by MPEG-7 to evaluate the retrieval performance [37]. This measure combines the precision and recall measure to obtain a single objective value. To use this measure, the queries and ground truth sets are chosen manually. For each query a set of ground truth items that are most relevant to the query are identified. The ground truth items are not ordered in any way. A good descriptor is expected to retrieve all the ground truth items for a given query image.

Let the number of ground truth items for a given query q be $NG(q)$, first $K(q)$ items are retrieved for query q by the tested retrieval system, a suitable $K(q)$ is determined by $K(q) = min(4 \cdot NG(q), 2 \cdot GTM)$ and GTM is the maximum of $NG(q)$ for all queries of the database. For each ground truth image $item_k$ that is retrieved in the top $K(q)$ retrieval, a rank value $Rank(item_k)$ is attached. The $Rank(item_k)$ is the retrieval rank of the ground truth image. The rank of the first retrieved item is one and the last is $K(q)$. A rank of $(K(q)+1)$ is assigned to each of the ground truth items, which are not in the first $K(q)$ retrievals.

Then the normalized modified retrieval rank (NMRR) is defined by

$$NMRR(q) = \frac{\sum_{k=1}^{NG(q)} \frac{Rank(item_k)}{NG(q)} - 0.5 - \frac{NG(q)}{2}}{K(q) + 0.5 - 0.5 \cdot NG(q)} \tag{1.24}$$

The NMRR is in the range of $[0, 1]$ and smaller values represent a better retrieval performance. For example, for the query q, the ground truth items are $I_{q1}, I_{q2}, \cdots I_{q10}$, that is $NG(q) = 10$. The ideal result is that the top 10 retrievals are these ground truth items $I_{q1}, I_{q2}, \cdots I_{q10}$, then their retrieval ranks are $Rank(I_{q1}) = 1$, $Rank(I_{q2}) = 2$, $\cdots$,$Rank(I_{q10}) = 10$, respectively. So the numerator in Eq. 1.24: $\sum_{k=1}^{10} \frac{Rank(I_{qk})}{10} - 0.5 - \frac{10}{2} = 0$.

ANMRR is defined as the average NMRR over a range of queries, and is given by

$$ANMRR = \frac{1}{NQ} \sum_{q=1}^{NQ} NMRR(q)$$

where NQ is the number of query items.

1.5 Conclusion

In this chapter, some fundamental concepts used in shape-based image description and retrieval have been introduced. First we have proposed an approach of contour equidistance normalization and discussed some remarkable problems about this normalization. We have compared the normalized precision and cost under different contour normalization methods. Some common shape parameters have been introduced: but these parameters representing certain shape geometric features can only discriminate shapes with large differences, they are not suitable to be standalone shape descriptors. We have introduced the usually used distance and similarity measure. Since the perception of human does not hold in a metric distance, a simple elastic matching method in non-metric space has also been presented in this section. Finally, the evaluation methods of retrieval system performance were defined.

All these presented fundamental concepts cover shape normalization, simple geometric features, similarity measure and evaluation methods. All of them construct the base of shape-based feature extraction and similarity measure. In the next chapter, we will focus on the existing algorithms of shape-based feature extraction and representation.

Chapter 2

Shape-based feature extraction and representation

2.1 Introduction

Pattern recognition is the ultimate goal of most computer vision research. Shape feature extraction and representation are the bases of object recognition. It is also a research domain which plays an important role in many applications ranging from image analysis and pattern recognition, to computer graphics and computer animation. The feature extraction stage produces a representation of the content that is useful for shape matching. Usually the shape representation is kept as compact as possible for the purposes of efficient storage and retrieval and it integrates perceptual features that allow the human brain to discriminate between shapes. Efficient shape features must present some essential properties such as:

- identifiability: shapes which are found perceptually similar by human have the same feature different from the others.
- translation, rotation and scale invariance: the location, rotation and scaling changing of the shape must not affect the extracted features.
- affine invariance: the affine transform performs a linear mapping from 2D coordinates to other 2D coordinates that preserves the "straightness" and "parallelism" of lines. Affine transform can be constructed using sequences of translations, scales, flips, rotations and shears. The extracted features must be as invariant as possible with affine transforms.
- noise resistance: features must be as robust as possible against noise, i.e., they must be the same whichever be the strength of the noise in a give range that affects the pattern.
- occultation invariance: when some parts of a shape are occulted by other objects, the feature of the remaining part must not change compared to the original shape.
- statistically independent: two features must be statistically independent. This represents compactness of the representation.
- reliable: as long as one deals with the same pattern, the extracted features must remain the same.

In general, shape descriptor is some set of numbers that are produced to describe a given shape feature. A descriptor attempts to quantify shape in ways that agree with human intuition (or task-specific requirements). Good retrieval accuracy requires a shape descriptor to be able to effectively find perceptually similar shapes from a database. Usually, the descriptors are in the form of a vector. Shape descriptors should meet the following requirements:

- the descriptors should be as complete as possible to represent the content of the information items.
- the descriptors should be represented and stored compactly. The size of descriptor vector must not be too large.
- the computation of distance between descriptors should be simple; otherwise the execution time would be too long.

So an efficient shape descriptor should have the following properties: accessibility, large scope, uniqueness and stability .

- accessibility describes how easy (or difficult) it is to compute a shape descriptor in terms of memory requirements and computation time.
- scope indicates the class of shapes that can be described by the method.
- uniqueness describes whether a one-to-one mapping exists between shapes and shape descriptors.
- stability describes how stable a shape descriptor is to "small" changes in shape.

Shape feature extraction and representation plays an important role in the following categories of applications:

- shape retrieval: searching for all shapes in a typically large database of shapes that are similar to a query shape. Usually all shapes within a given distance from the query are determined or the first few shapes that have the smallest distance.
- shape recognition and classification: determining whether a given shape matches a model sufficiently, or which of representative class is the most similar.
- shape alignment and registration: transforming or translating one shape so that it best matches another shape, in whole or in part.
- shape approximation and simplification: constructing a shape of fewer elements (points, segments, triangles, etc.), that is still similar to the original.

Many shape description and similarity measurement techniques have been developed in the past. A number of new techniques have been proposed in recent years. There are 3 main different classification methods as follows:

- Contour-based methods and region-based methods [2]. This is the most common and general classification and it is proposed by MPEG-7. It is based on the use of shape boundary points as opposed to shape interior points. Under each class, different methods are further divided into structural approaches and global approaches. This sub-class is based on whether the shape is represented as a whole or represented by segments/sections (primitives).

- Space domain and transform domain. Methods in space domain match shapes on point (or point feature) basis, while feature domain techniques match shapes on feature (vector) basis.

- Information preserving (IP) and non-information preserving (NIP). IP methods allow an accurate reconstruction of a shape from its descriptor, while NIP methods are only capable of partial ambiguous reconstruction. For object recognition purpose, IP is not a requirement.

Various algorithms and methods are documented in a vast literatures. In this thesis, for sake of application conveniences, we reclassify them according to the processing methods. The whole hierarchy of the classification is shown in Figure 2.1.

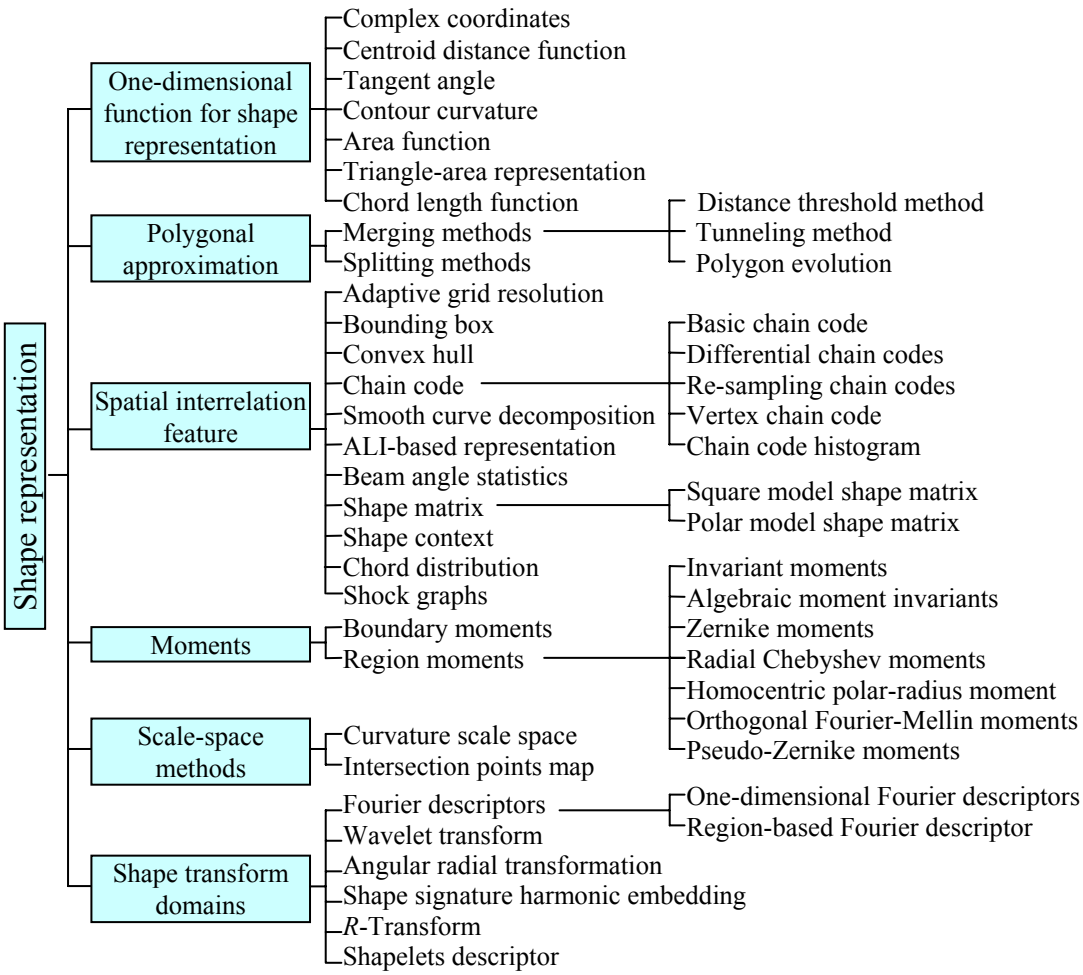


Figure 2.1: An overview of shape description techniques

Without being complete, in the following chapter, we will describe and group a number of these methods together.

2.2 One-dimensional function for shape representation

The one-dimensional function which is derived from shape boundary coordinates is also often called shape signature [38, 13]. The shape signature usually captures the perceptual feature of the shape [39]. Complex coordinates, Centroid distance function, Tangent angle (Turning angles), Curvature function, Area function, Triangle-area representation and Chord length function are the commonly used shape signatures.

Shape signature can describe a shape all alone; it is also often used as a preprocessing to other feature extraction algorithms, for example, Fourier descriptors, wavelet description. In this section, the shape signatures are introduced.

2.2.1 Complex coordinates

A complex coordinates function is simply the complex number generated from the coordinates of boundary points, $P_n(x(n), y(n))$, $n \in [1, N]$:

$$z(n) = [x(n) - g_x] + i[y(n) - g_y]$$

where (g_x, g_y) is the centroid of the shape, given by Eq. 1.9.

2.2.2 Centroid distance function

The centroid distance function is expressed by the distance of the boundary points from the centroid (g_x, g_y) (Eq. 1.9) of a shape

$$r(n) = [(x(n) - g_x)^2 + (y(n) - g_y)^2]^{1/2}$$

Due to the subtraction of centroid, which represents the position of the shape, from boundary coordinates, both complex coordinates and centroid distance representation are invariant to translation.

2.2.3 Tangent angle

The tangent angle function at a point $P_n(x(n), y(n))$ is defined by a tangential direction of a contour [40]:

$$\theta(n) = \theta_n = \arctan \frac{y(n) - y(n-w)}{x(n) - x(n-w)}$$

since every contour is a digital curve; w is a small window to calculate $\theta(n)$ more accurately.

Tangent angle function has two problems. One is noise sensitivity. To decrease the effect of noise, a contour is filtered by a low-pass filter with appropriate bandwidth before calculating the tangent angle function. The other is discontinuity, due to the fact that the tangent angle function assumes values in a range of length 2π, usually in the interval of $[-\pi, \pi]$ or $[0, 2\pi]$. Therefore θ_n in general contains discontinuities of size 2π. To overcome the discontinuity problem, with an arbitrary starting point, the cumulative angular function φ_n is defined as the angle differences between the tangent at any point P_n along the curve and the tangent at the starting point P_0 [41, 42]:

$$\varphi(n) = [\theta(n) - \theta(0)]$$

In order to be in accordance with human intuition that a circle is "shapeless", assume $t = 2\pi n/N$, then $\varphi(n) = \varphi(tN/2\pi)$. A periodic function is termed as the cumulative angular deviant function $\psi(t)$ and is defined as

$$\psi(t) = \varphi(\frac{N}{2\pi}t) - t \quad t \in [0, 2\pi]$$

N is the total number of contour points.

In [43], the authors proposed a method based on tangent angle. It is called tangent space representation. A digital curve C simplified by polygon evolution is represented in the tangent space by the graph of a step function, where the x-axis represents the arc length coordinates of points in C and the y-axis represents the direction of the line segments in the decomposition of C. For example, figure 2.2 shows a digital curve and its step function representation in the tangent space.

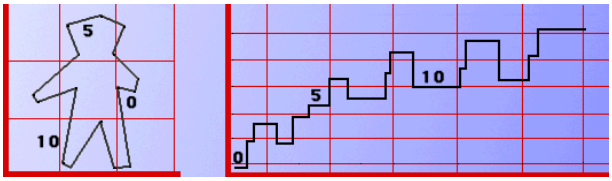


Figure 2.2: Digital curve and its step function representation in the tangent space

2.2.4 Contour curvature

Curvature is a very important boundary feature for human to judge similarity between shapes. It also has salient perceptual characteristics and has proven to be very useful for shape recognition [44]. In order to use *K(n)* for shape representation, we quote the function of curvature, *K(n)*, from [45, 46] as:

$$K(n) = \frac{\dot{x}(n)\ddot{y}(n) - \dot{y}(n)\ddot{x}(n)}{(\dot{x}(n)^2 + \dot{y}(n)^2)^{3/2}} \tag{2.1}$$

Therefore, it is possible to compute the curvature of a planar curve from its parametric representation. If n is the normalized arc-length parameter s, then Eq. 2.1 can be written as:

$$K(s) = \dot{x}(s)\ddot{y}(s) - \dot{y}(s)\ddot{x}(s) \tag{2.2}$$

As given in Eq. 2.2, the curvature function is computed only from parametric derivatives, and, therefore, it is invariant under rotations and translations. However, the curvature measure is scale dependent, i.e., inversely proportional to the scale. A possible way to achieve scale independence is to normalize this measure by the mean absolute curvature, i.e.,

$$K'(s) = \frac{K(s)}{\frac{1}{N}\sum_{s=1}^{N}|K(s)|}$$

where N is the number of points on the normalized contour.

When the size of the curve is an important discriminative feature, the curvature should be used without the normalization; otherwise, for the purpose of scale-invariant shape analysis, the normalization should be performed.

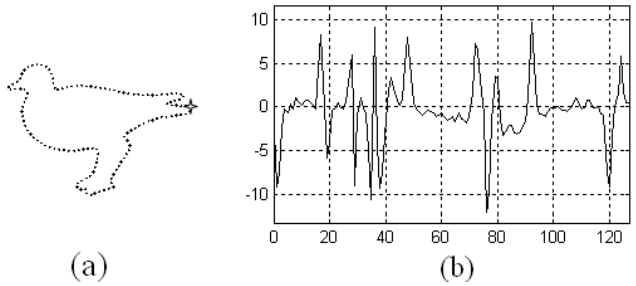
An approximate arc-length parametrization based on the centripetal method is given by the following [46]:

Let $P = \sum_{n=1}^{N} d_n$ be the perimeter of the curve and $L = \sum_{n=1}^{N} \sqrt{d_n}$, where d_n is the length of the chord between points p_n and p_{n+1}, *n=1, 2, . . . , N-1.* The approximate arc-length parametrization relations:

$s_1 = 0$;

$s_k = s_{k-1} + \frac{P\sqrt{d_{k-1}}}{L}, k = 2, 3, \ldots, N.$

Starting from an arbitrary point and following the contour clockwise, we compute the curvature at each interpolated point using Eq. 2.2. Convex and concave vertices will imply negative and positive values, respectively (the opposite is verified for counterclockwise sense). Figure 2.3 is an example of curvature function. Clearly, as a descriptor, the curvature function can distinguish different shapes.

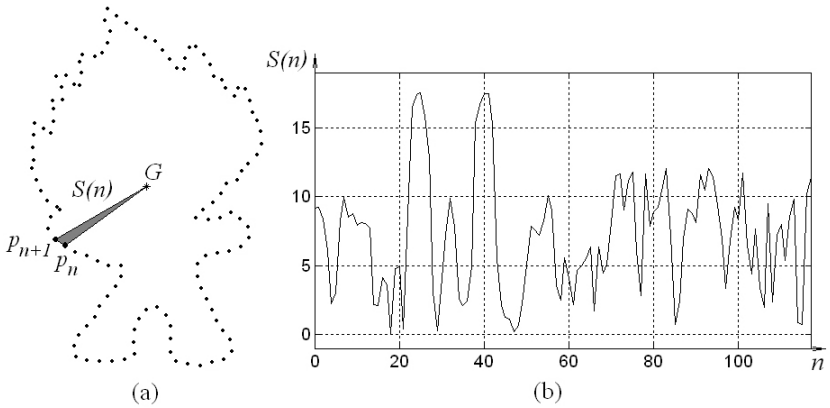


(a) Contours normalized to 128 points; the dots marked star are the starting points on the contours; (b) curvature functions; the curvature is computed clockwise.

Figure 2.3: Curvature function

2.2.5 Area function

When the boundary points change along the shape boundary, the area of the triangle formed by two successive boundary points and the center of gravity also changes. This forms an area function which can be exploited as shape representation. Figure 2.4 shows an example. Let $S(n)$ be the area between the successive boundary points P_n, P_{n+1} and center of gravity G.



(a) Original contour; (b) the area function of (a).

Figure 2.4: Area function

The area function is linear under affine transform. However, this linearity only works for shape sampled at its same vertices.

2.2.6 Triangle-area representation

The triangle-area representation (TAR) signature is computed from the area of the triangles formed by the points on the shape boundary [47, 48]. The curvature of the contour point (x_n, y_n) is measured using the TAR as follows.

For each three consecutive points $P_{n-t_s}(x_{n-t_s}, y_{n-t_s})$, $P_n(x_n, y_n)$, and $P_{n+t_s}(x_{n+t_s}, y_{n+t_s})$, where $n \in [1, N]$ and $t_s \in [1, N/2-1]$, N is even. The signed area of the triangle formed by these points is given by:

$$TAR(n, t_s) = \frac{1}{2} \begin{vmatrix} x_{n-t_s} & y_{n-t_s} & 1 \\ x_n & y_n & 1 \\ x_{n+t_s} & y_{n+t_s} & 1 \end{vmatrix} \tag{2.3}$$

When the contour is traversed in counter clockwise direction, positive, negative and zero values of TAR mean convex, concave and straight-line points, respectively. Figure 2.5 demonstrates these three types of the triangle areas and the complete TAR signature for the hammer shape.

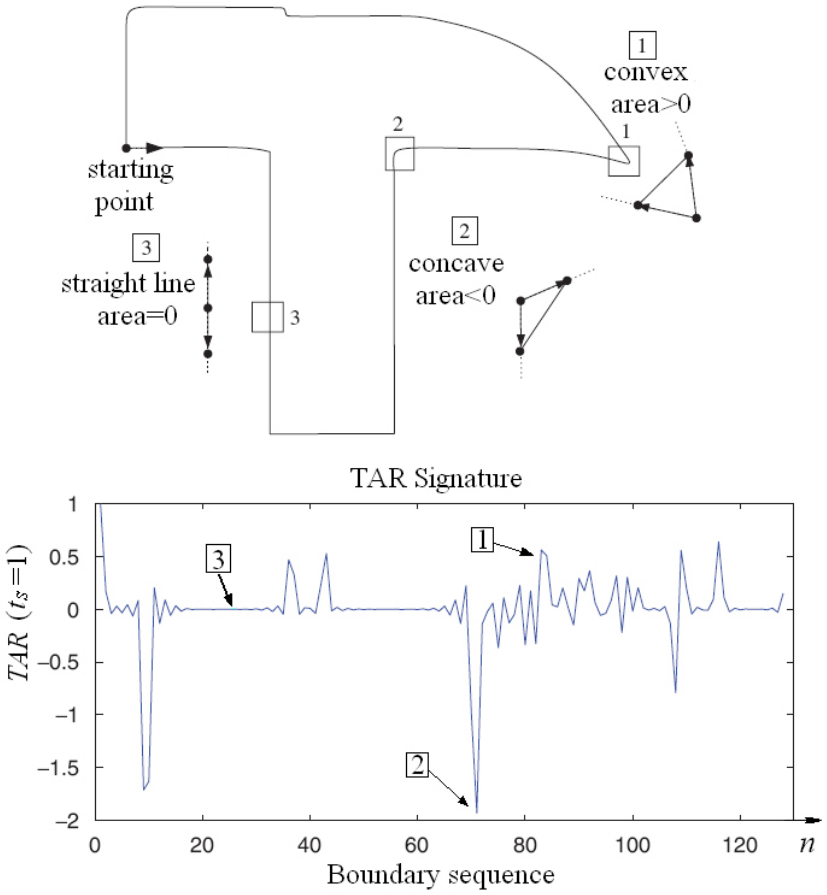
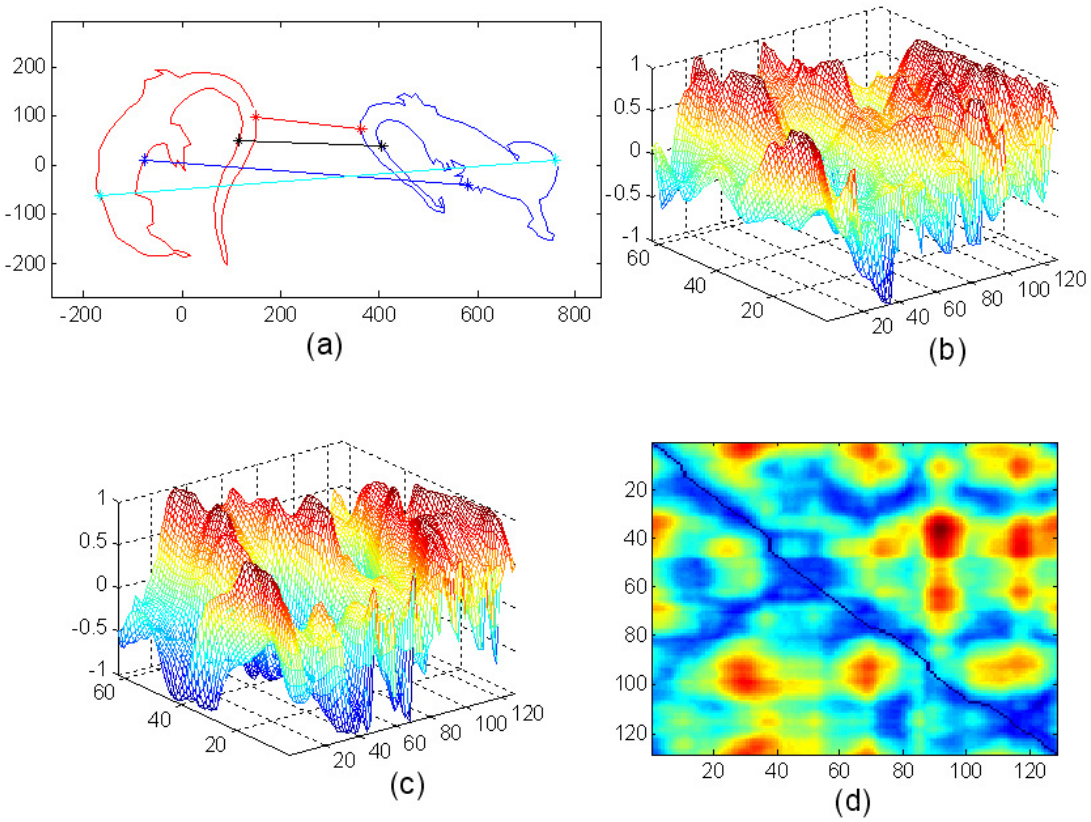


Figure 2.5: Three different types of the triangle-area values and the TAR signature for the hammer shape

By increasing the length of the triangle sides, i.e., considering farther points, the function of Eq. 2.3 will represent longer variations along the contour. The TARs with different triangle sides

can be regarded as different scale space functions. The total TARs, $t_s \in [1, N/2-1]$, compose a multi-scale space TAR.

Figure 2.6 shows the multi-scale space TAR function and its dynamic space warping (DSW) matching (cf. Subsection 1.3.2). In (a), the correspondent points on the model contour to these on the query contour are consistent with human perception after DSW matching.



(a) The query contour (left) and the model contour (right); (b) the multi-scale space TAR function of the model contour; (c) the multi-scale space TAR function of the query contour; (d) dynamic space warping (DSW) matching table of the multi-scale space TAR functions (b) and (c).

Figure 2.6: Dynamic space warping (DSW) matching

In [48], authors show that the multi-scale space TAR is relatively invariant to the affine transform and robust to non-rigid transform. The computation complexity of the TAR stage is $O(N^2)$.

2.2.7 Chord length function

The chord length function is derived from shape boundary without using any reference point. For each boundary point p, its chord length function is the shortest distance between p and another boundary point p' such that line pp' is perpendicular to the tangent vector at p [38].

The chord length function is invariant to translation and it overcomes the biased reference point (which means the centroid is often biased by boundary noise or defections) problems. However, it is very sensitive to noise, there may be drastic burst in the signature of even smoothed shape boundary.

2.2.8 Discussions

A shape signature represents a shape by a 1-D function derived from shape contour. To obtain the translation invariant property, they are usually defined by relative values. To obtain the scale invariant property, normalization is necessary. In order to compensate for orientation changes, shift matching is needed to find the best matching between two shapes. Having regard to occultation, Tangent angle, Contour curvature and Triangle-area representation have invariance property. In addition, shape signatures are computationally simple.

Shape signatures are sensitive to noise, and slight changes in the boundary can cause large errors in matching. Therefore, it is undesirable to directly describe shape using a shape signature. Further processing is necessary to increase its robustness and reduce the matching load. For example, a shape signature can be simplified by quantizing the signature into a signature histogram, which is rotationally invariant.

2.3 Polygonal approximation

Polygonal approximation can be set to ignore the minor variations along the edge, and instead capture the overall shape. This is useful because it reduces the effects of discrete pixelization of the contour. In general, there are two methods to realize it. One is merging, the other is splitting [49].

2.3.1 Merging methods

Merging methods add successive pixels to a line segment if each new pixel that is added doesn't cause the segment to deviate too much from a straight line.

Distance threshold method

Choose one point as a starting point, on the contour, for each new point that we add, let a line go from the starting point to this new point. Then, we compute the squared error for every point along the segment/line. If the error exceeds some threshold, we keep the line from the start point to the previous point and start a new line.

In practice, the most of practical error measures in use are based on distance between vertices of the input curve and the approximation linear segments. The distance $d_k(i,j)$ from curve vertex $P_k = (x_k, y_k)$ to the corresponding approximation linear segments (P_i, P_j) is defined as follows (cf. Figure 2.7):

$$d_k(i,j) = \frac{|(x_j - x_i)(y_i - y_k) - (x_i - x_k)(y_j - y_i)|}{\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}}$$

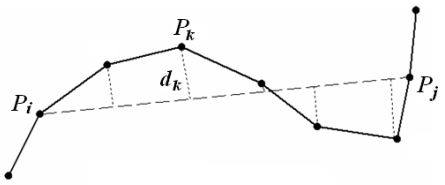


Figure 2.7: Illustration of the distance from a point on the boundary to a linear segment

Tunneling method

If we have thick boundaries rather than single-pixel thick ones, we can still use a similar approach called tunneling. Imagine that we're trying to lay straight rods along a curved tunnel, and that we want to use as few as possible. We can start at one point and lay as long a straight rod as possible. Eventually, the curvature of the "tunnel" won't let us go any further, so we lay another rod and another until we reach the end.

Both the distance threshold and tunneling methods can do polygonal approximation efficiently. However, the great disadvantage is that the position of starting point will affect greatly the approximate polygon.

Polygon evolution

The basic idea of polygons evolution in [50] is very simple: in every evolution step, a pair of consecutive line segments (the line segment is the line between two consecutive vertices) s_1,s_2 is substituted with a single line segment joining the endpoints of s_1 and s_2.

The key property of this evolution is the order of the substitution. The substitution is done according to a relevance measure K given by

$$K(s_1, s_2) = \frac{\beta(s_1, s_2) l(s_1) l(s_2)}{l(s_1) + l(s_2)},$$

where $\beta(s_1, s_2)$ is the turn angle at the common vertex of segments s_1, s_2 and $l(\alpha)$ is the length of α, $\alpha = s_1 \, or \, s_2$, normalized with respect to the total length of a polygonal curve. The evolution algorithm is assuming that vertices which are surrounded by segments with a high value of $K(s_1, s_2)$ are important while those with a low value are not. Figure 2.8 is an example.

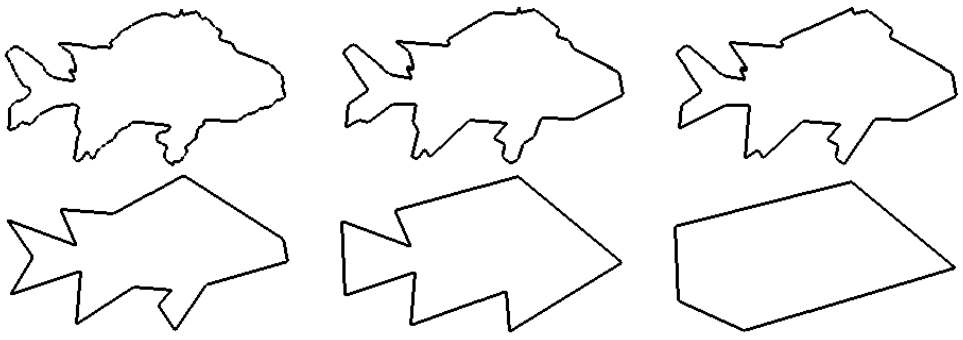


Figure 2.8: A few stages of polygon evolution according to a relevant measure

The curve evolution method achieves the task of shape simplification, i.e., the process of evolution compares the significance of vertices of the contour based on a relevance measure. Since any digital curve can be regarded as a polygon without loss of information (with possibly a large number of vertices), it is sufficient to study evolutions of polygonal shapes for shape feature extraction.

2.3.2 Splitting methods

Splitting methods work by first drawing a line from one point on the boundary to another. Then, we compute the perpendicular distance from each point along the boundary segment to the line. If this exceeds some threshold, we break the line at the point of greatest distance. We then repeat the process recursively for each of the two new lines until we don't need to break any more. See Figure 2.9 for an example.

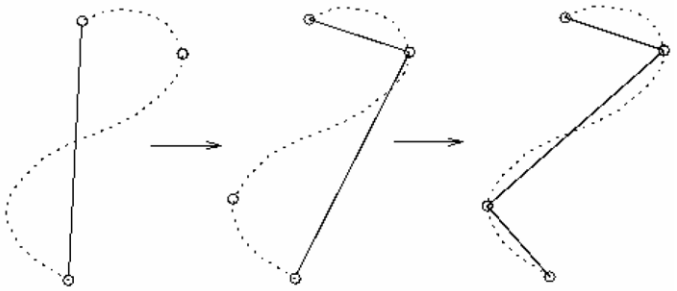


Figure 2.9: Splitting methods for polygonal approximation

This is sometimes known as the "fit and split" algorithm. For a closed contour, we can find the two points that lie farthest apart and fit two lines between them, one for one side and one for the other. Then, we can apply the recursive splitting procedure to each side.

2.3.3 Discussions

Polygonal approximation technique can be used as a simple method for contour representation and description. The polygon approximation have some interesting properties:

- it leads to simplification of shape complexity with no blurring effects.
- it leads to noise elimination.
- although irrelevant features vanish after polygonal approximation, there is no dislocation of relevant features.
- the remaining vertices on a contour do not change their positions after polygonal approximation.

Polygonal approximation technique can also be used as preprocessing method for further extracting features from a shape.

2.4 Spatial interrelation feature

Spatial interrelation feature describes the region or the contour of shape by the relation of their pixels or curves. In general, the representation is done by using their geometric features: length, curvature, relative orientation and location, area, distance and so on.

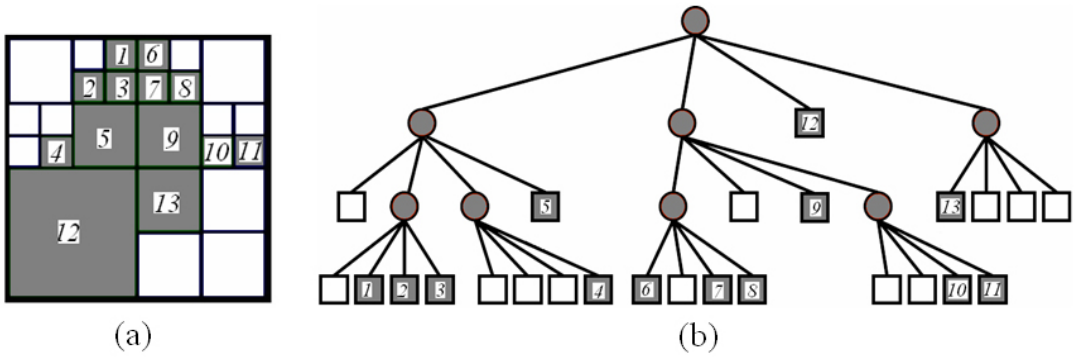
2.4.1 Adaptive grid resolution

The adaptive grid resolution (AGR) was proposed by [51]. In the AGR, a square grid that is just big enough to cover the entire shape is overlaid on a shape. A resolution of the grid cells varies from one portion to another according to the content of the portion of the shape. On the borders or the detail portion on the shape, the higher resolution, i.e. the smaller grid cells, are applied; on the other hand, in the coarse regions of the shape, lower resolution, i.e. the bigger grid cells, are applied.

To guarantee rotation invariance, it needs to convert an arbitrarily oriented shape into a unique common orientation. First, find the major axis of the shape. The major axis is the straight line segment joining the two points P_1 and P_2 on the boundary farthest away from each other. Then we rotate the shape so that its major axis is parallel to the x-axis. This orientation is still not unique as there are two possibilities: P_1 can be on the left or on the right. This problem is solved by computing the centroid of the polygon and making sure that the centroid is below the major axis, thus guaranteeing a unique orientation.

Let us now consider scale and translation invariance. We define the bounding rectangle (BR) of a shape as the rectangle with sides parallel to the x and y axes just large enough to cover the entire shape (after rotation). Note that the width of the BR is equal to the length of the major axis. To achieve scale invariance, we proportionally scale all shapes so that their BRs have the same fixed width (pixels).

The method of computation of the AGR representation of a shape applies quad-tree decomposition on the bitmap representation of the shape. The decomposition is based on successive subdivision of the bitmap into four equal-size quadrants. If a bitmap-quadrant does not consist entirely of part of shape, it is recursively subdivided into smaller and smaller quadrants until we reach bitmap-quadrants, i.e., termination condition of the recursion is that the resolution reaches that one predefined. Figure 2.10(a) is an example of AGR.



(a) Adaptive Grid Resolution (AGR) image; (b) quad-tree decomposition of AGR.

Figure 2.10: Adaptive resolution representations

To represent the AGR image, in [51], quad-tree method is applied. Each node in the quad-

tree covers a square region of the bitmap. The level of the node in the quad-tree determines the size of the square. The internal nodes (shown by gray circles) represent "partially covered" regions; the leaf nodes shown by white boxes represent regions with all $0s$ while the leaf nodes shown by black boxes represent regions with all $1s$. The "all $1s$" regions are used to represent the shape, Figure 2.10(b). Each rectangle can be described by 3 numbers: its center $C = (C_x, C_y)$ and its size (i.e. side length) S. So each shape can be mapped to a point in $3n$-dimensional space (n is the number of the rectangles occupied by the shape region).

Due to the fact that the normalization before computing AGR, AGR representation is invariant under rotation, scaling and translation. It is also computationally simple.

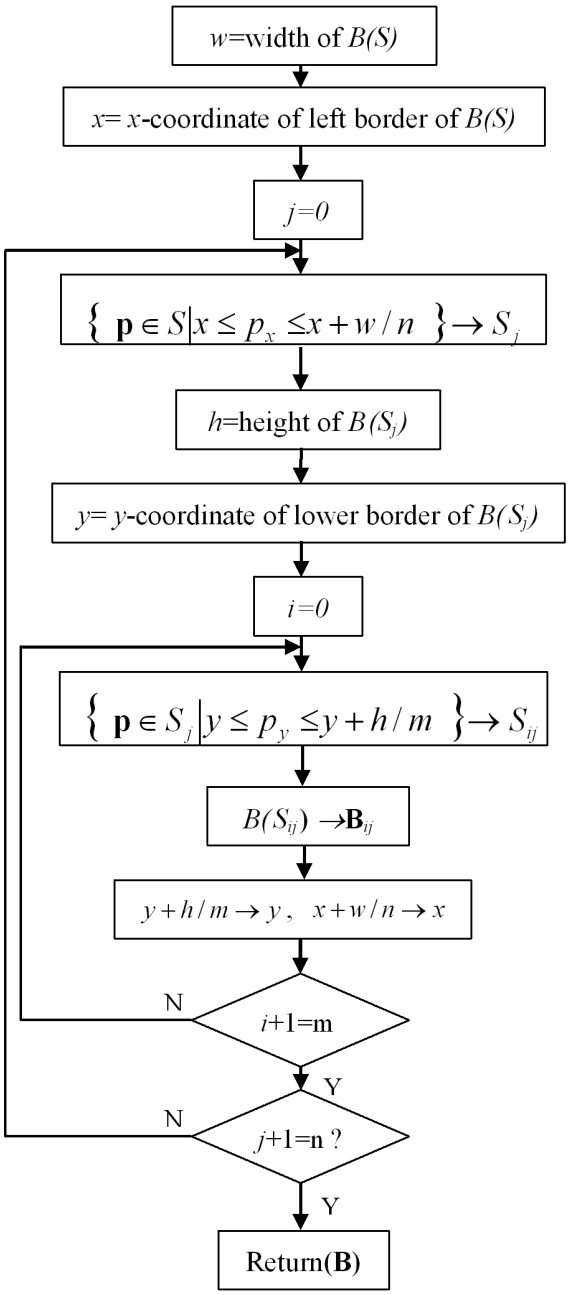


Figure 2.11: Flowchart of shape divided by bounding box

2.4.2 Bounding box

Bounding box computes homeomorphisms between 2D lattices and its shapes. Unlike many other methods, this mapping is not restricted to simply connected shapes but applies to arbitrary topologies [52].

To make bounding box representation invariant to rotation, a shape should be normalized by the same method as for AGR (Subsection 2.4.1) before further computation. After the normalization, a shape S is a set of L pixels, $S = \left\{p_k \in R^2 | k = 1, 2, \cdots, L\right\}$ and also write $|S| = L$. The minimum bounding rectangle or bounding box of S is denoted by $B(S)$; its width and height, are called w and h, respectively.

Figure 2.11 shows the algorithm flowchart based on bounding box that divides a shape S into $m\,(row) \times n\,(column)$ parts. The output $\mathbf{B}$ is a set of bounding boxes.

An illustration of this procedure and its result is shown in Figure 2.12.

(a) Compute the bounding box $B(S)$ of a pixel set S; (b) subdivide S into n vertical slices; (c) compute the bounding box $B(S_j)$ of each resulting pixel set S_j, where $j = 1, 2, \cdots, n$; (d) subdivide each $B(S_j)$ into m horizontal slices; (e) compute the bounding box $B(S_{ij})$ of each resulting pixel set S_{ij}, where $i = 1, 2, \cdots, m$.

Figure 2.12: The five steps of bounding box splitting

To represent each bounding box, one method is that partial points of the set of bounding boxes are sampled. Figure 2.13 shows an example.

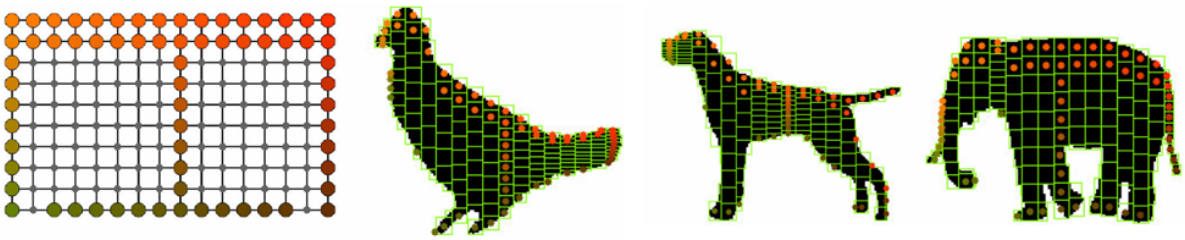


Figure 2.13: A sample points on lattice and examples of how it is mapped onto different shapes

If $\mathbf{v} = (v_x, v_y)^T$ denotes the location of the bottom left corner of the initial bounding box of S, and $\mathbf{u}_{ij} = (u_x^{ij}, u_y^{ij})$ denotes the center of sample box $\mathbf{B}_{ij}$, then the coordinates

$$\begin{pmatrix} \mu_x^{ij} \\ \mu_y^{ij} \end{pmatrix} = \begin{pmatrix} \left(u_x^{ij} - v_x\right)/w \\ \left(u_y^{ij} - v_y\right)/h \end{pmatrix}$$

provide a scale invariant representation of S. Sampling k points of an $m \times n$ lattice therefore allows to represent S as a vector

$$r = \left[\mu_x^{i(1)j(1)}, \mu_y^{i(1)j(1)}, \cdots, \mu_x^{i(k)j(k)}, \mu_y^{i(k)j(k)}\right]$$

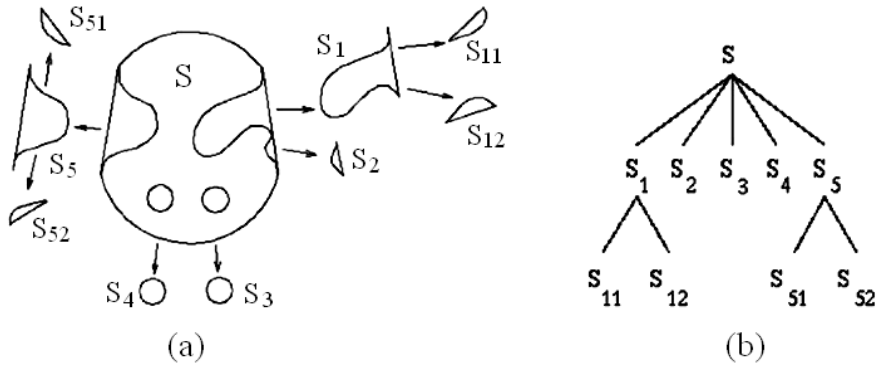
where $i(\alpha) < i(\beta)$ if $\alpha < \beta$ and likewise for the index j.

Bounding box representation is a simple computational geometry approach to compute homeomorphisms between shapes and lattices. It is storage and time efficient. It is invariant to rotation, scaling and translation and also robust against noisy shape boundaries.

2.4.3 Convex hull

The approach is that the shape is represented by a serie of convex hulls. The convex region has be defined in Sebsection 1.2.8. The convex hull H of a region is its smallest convex region including it. In other words, for a region S, the convex hull $conv(S)$ is defined as the smallest convex set in R^2 containing S. In order to decrease the effect of noise, common practice is to first smooth a boundary prior to partitioning.

The representation of the shape may be obtained by a recursive process which results in a concavity tree. See Figure 2.14. Each concavity can be described by its area, chord (the line connects the cut of the concavity) length, maximum curvature, distance from maximum curvature point to the chord. The matching between shapes becomes a string or a graph matching.



(a) Convex hull and its concavities; (b) concavity tree representation of convex hull.

Figure 2.14: Illustrates recursive process of convex hull

Convex hull representation has a high storage efficiency. It is invariant to rotation, scaling and translation and also robust against noisy shape boundaries (after filtering). However, extracting the robust convex hulls from the shape is where the shoe pinches. [53, 54] and [55] gave the boundary tracing method and morphological methods to achieve convex hulls respectively.

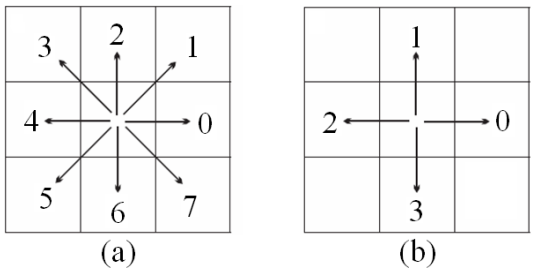
2.4.4 Chain code

Chain code is a common approach for representing different rasterized shapes as line-drawings, planar curves, or contours. Chain code describes an object by a sequence of unit-size line segments with a given orientation [8]. Chain code can be viewed as a connected sequence of straight-line segments with specified lengths and directions [56].

Basic chain code

Freeman [57] first introduced a chain code that describes the movement along a digital curve or a sequence of border pixels by using so-called 8-connectivity or 4-connectivity. The direction of each movement is encoded by the numbering scheme $\{i|i = 0, 1, 2, \cdots, 7\}$ or $\{i|i = 0, 1, 2, 3\}$

denoting a counter-clockwise angle of $45° \times i$ or $90° \times i$ regarding the positive x-axis, as shown in Figure 2.15.



(a) Chain code in eight directions (8-connectivity); (b) chain code in four directions (4-connectivity).

Figure 2.15: Basic chain code direction

By encoding relative, rather than absolute position of the contour, the basic chain code is translation invariant. We can match boundaries by comparing their chain codes, but with the two main problems: 1) it is very sensitive to noise; 2) it is not rotationally invariant. To solve these problems, differential chain codes (DCC) and resampling chain codes (RCC) were proposed.

Differential chain codes (DCC) is encoding differences in the successive directions. This can be computed by subtracting each element of the chain code from the previous one and taking the result modulo n, where n is the connectivity. This differencing allows us to rotate the object in 90-degree increments and still compare the objects, but it doesn't get around the inherent sensitivity of chain codes to rotation on the discrete pixel grid.

Re-sampling chain codes (RCC) consists in re-sampling the boundary onto a coarser grid and then computing the chain codes of this coarser representation. This smoothes out small variations and noise but can help compensate for differences in chain-code length due to the pixel grid.

Vertex chain code (VCC)

To improve chain code efficiency, in [56] the authors proposed a chain code for shape representation according to vertex chain code (VCC). An element of the VCC indicates the number of cell vertices, which are in touch with the bounding contour of the shape in that element's position. Only three elements "1", "2" and "3" can be used to represent the bounding contour of a shape composed of pixels in the rectangular grid. Figure 2.16 shows the elements of the VCC to represent a shape.

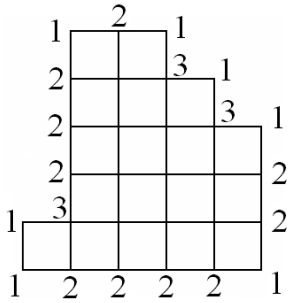


Figure 2.16: Vertex chain code

Chain code histogram (CCH)

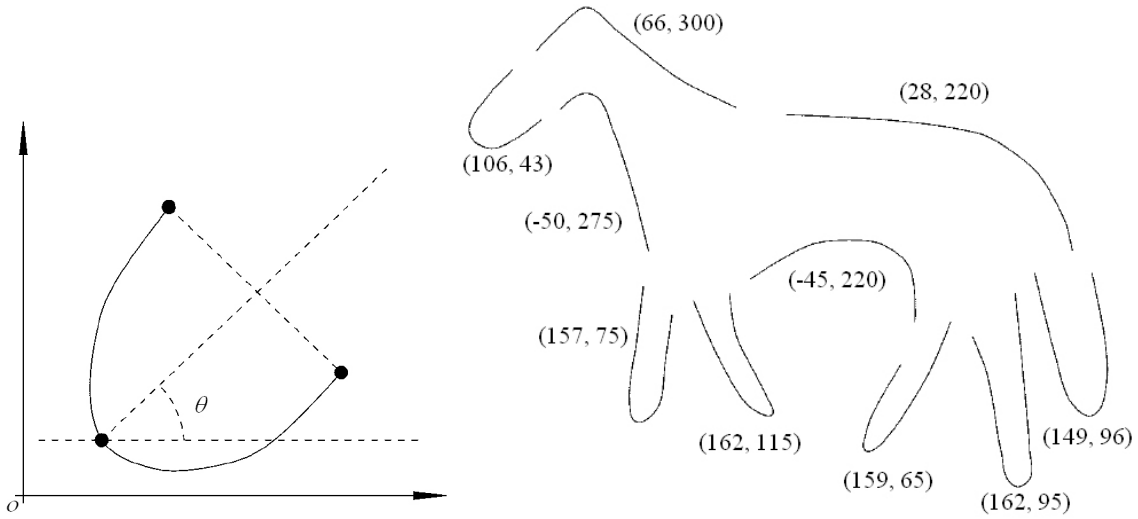
Iivarinen and Visa derive a chain code histogram (CCH) for object recognition [58]. The CCH is computed as $h_i = \sharp\{i \in M,\ M \text{ is the range of chain code}\}$, $\sharp\{\alpha\}$ denotes getting the number of the value α.

The CCH reflects the probabilities of different directions present in a contour.

If the chain code is used for matching it must be independent of the choice of the starting pixel in the sequence. The chain code usually has high dimensions and is sensitive to noise and any distortion. So, except the CCH, the other chain code approaches are often used as contour representations, but is not as contour attributes.

2.4.5 Smooth curve decomposition

In [5], the authors proposed smooth curve decomposition as shape descriptor. The segment between the curvature zero-crossing points from a Gaussian smoothed boundary are used to obtain primitives, called tokens. The feature for each token is its maximum curvature and its orientation. In Figure 2.17, the first number in the parentheses is its maximum curvature and the second is its orientation.



(a) θ is the orientation of this token; (b) an example of smooth curve decomposition.

Figure 2.17: Smooth curve decomposition

The similarity between two tokens is measured by the weighted Euclidean distance. The shape similarity is measured according to a non-metric distance. Shape retrieval based on token representation has shown to be robust in the presence of partially occulted objects, translation, scaling and rotation.

2.4.6 Symbolic representation based on the axis of least inertia

In [59], a method of representing a shape in terms of multi-interval valued type data is proposed. The proposed shape representation scheme extracts symbolic features with reference to the axis of least inertia, which is unique to the shape. The axis of least inertia (ALI) of a shape is defined

as the line for which the integral of the square of the distances to points on the shape boundary is a minimum (cf. Subsection 1.2.2).

Once the ALI is calculated, each point on the shape curve is projected on to ALI. The two farthest projected points say $E1$ and $E2$ on ALI are chosen as the extreme points as shown in Figure 2.18. The Euclidean distance between these two extreme points defines the length of ALI. The length of ALI is divided uniformly by a fixed number n; the equidistant points are called feature points. At every feature point chosen, an imaginary line perpendicular to the ALI is drawn. It is interesting to note that these perpendicular lines may intersect the shape curve at several points. The length of each imaginary line in shape region is computed and the collection of these lengths in an ascending order defines the value of the feature at the respective feature point.

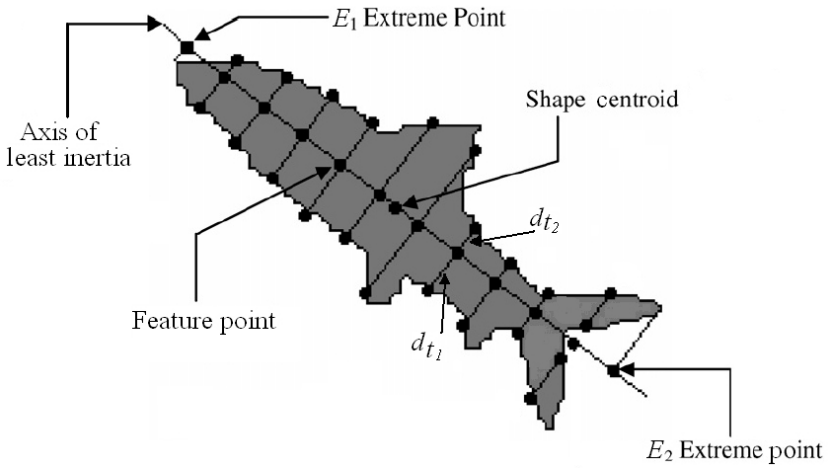


Figure 2.18: Symbolic features based axis of least inertia

Let S be a shape to be represented and n the number of feature points chosen on its ALI. Then the feature vector F representing the shape S, is in general of the form $F = [\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_t, ..., \mathbf{f}_n]$, where $\mathbf{f}_t = \{d_{t_1}, d_{t_2}, \cdots d_{t_k}\}$ for some $t_k \geq 1$.

The feature vector F representing the shape S is invariant to image transformations viz., uniform scaling, rotation, translation and flipping (reflection).

2.4.7 Beam angle statistics

Beam angle statistics (BAS) shape descriptor is based on the beams originated from a boundary point, which are defined as lines connecting that point with the rest of the points on the boundary [60].

Let B be the shape boundary. $B = \{P_1, P_2, \cdots, P_N\}$ is represented by a connected sequence of points, $P_i = (x_i, y_i)$, $i = 1, 2, \cdots, N$, where N is the number of boundary points. For each point P_i, the beam angle between the forward beam vector $V_{i+k} = \overrightarrow{P_i P_{i+k}}$ and backward beam vector $V_{i-k} = \overrightarrow{P_i P_{i-k}}$ in the k^{th} order neighborhood system, is then computed as (see Figure 2.19, k=5 for example)

$$C_k(i) = (\theta_{V_{i+k}} - \theta_{V_{i-k}})$$

where $\theta_{V_{i+k}} = \arctan \dfrac{y_{i+k} - y_i}{x_{i+k} - x_i}$, $\theta_{V_{i-k}} = \arctan \dfrac{y_{i-k} - y_i}{x_{i-k} - x_i}$

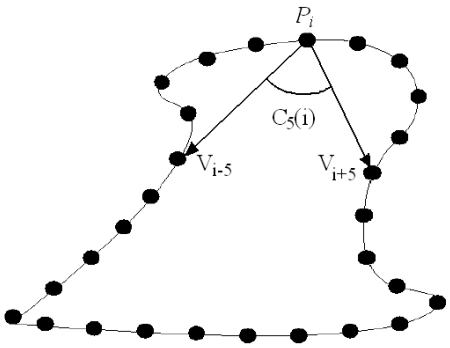
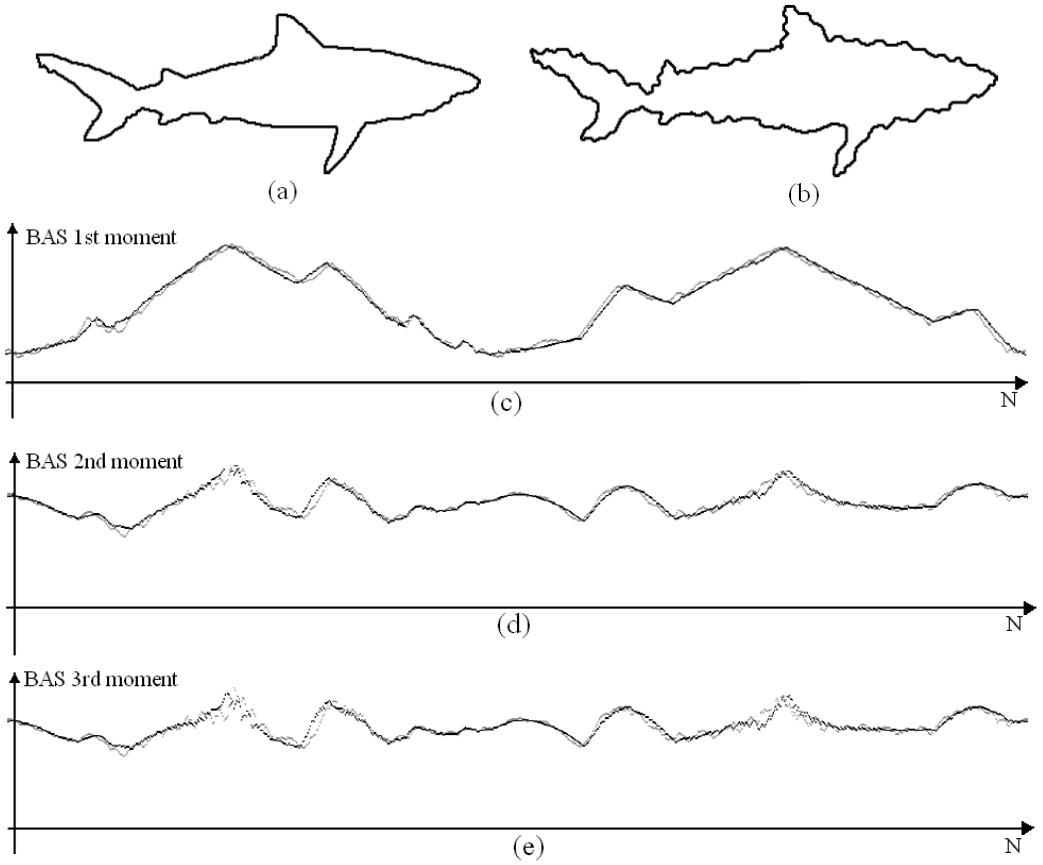


Figure 2.19: Beam angle at the neighborhood system 5 for a boundary point



(a) Original contour; (b) noisy contour; (c), (d) and (e) are the BAS plot 1^{st}, 2^{nd} and 3^{rd} moment, respectively.

Figure 2.20: The BAS descriptor for original and noisy contour

For each boundary point P_i of the contour, the beam angle $C_k(i)$ can be taken as a random variable with the probability density function $P(C_k(i))$. Therefore, beam angle statistics (BAS), may provide a compact representation for a shape descriptor. For this purpose, m^{th} moment of

the random variable $C_k(i)$ is defined as follows:

$$E[C^m(i)] = \sum_{k=1}^{(N/2)-1} C_k^m(i) \cdot P_k(C_k(i)) \quad m = 1, 2, \cdots$$

In the above formula E indicates the expected value. See Figure 2.20 as an example.

Beam angle statistics shape descriptor captures the perceptual information using the statistical information based on the beams of individual points. It gives globally discriminative features to each boundary point by using all other boundary points. BAS descriptor is also quite stable under distortions and is invariant to translation, rotation and scaling.

2.4.8 Shape matrix

Shape matrix descriptor is an $M \times N$ matrix to present a region shape. There are two basic modes of shape matrix: Square model [21] and Polar model [20].

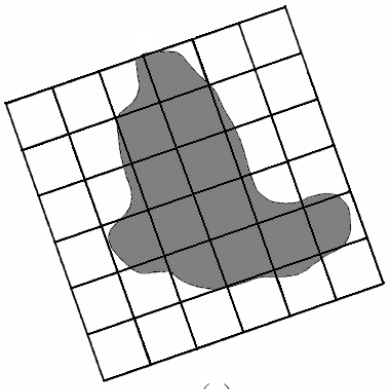
Square model shape matrix

Square model of shape matrix, also called grid descriptor [61, 21], is constructed by the following: for the shape S, construct a square centered on the center of gravity G of S; the size of each side is equal to $2L$, L is the maximum Euclidean distance from G to a point M on the boundary of the shape. Point M lies in the center of one side and GM is perpendicular to this side.

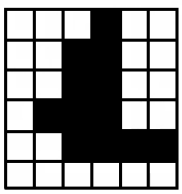
Divide the square into $N \times N$ subsquares and denote S_{kj}, $k, j = 1, \cdots, N$, the subsquares of the constructed grid. Define the shape matrix $SM = [B_{kj}]$,

$$B_{kj} = \begin{cases} 1 \Leftrightarrow & \mu(S_{kj} \cap S) \geq \mu(S_{kj})/2 \\ 0 & otherwise \end{cases}$$

where $\mu(F)$ is the area of the planar region F. Figure 2.21 shows an example of square model of shape matrix.



$$SM = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



(a) (b) (c)

(a) Original shape region; (b) square model shape matrix; (c) reconstruction of the shape region.

Figure 2.21: Square model shape matrix

For a shape with more than one maximum radius, it can be described by several shape matrices and the similarity distance is the minimum distance between these matrices. In [21], authors gave a method to choose the appropriate shape matrix dimension.

Polar model shape matrix

Polar model of shape matrix is constructed by the following steps. Let G be the center of gravity of the shape, and GA is the maximum radius of the shape. Using G as center, draw n circles with radii equally spaced. Starting from GA, and counterclockwise, draw radii that divide each circle into m equal arcs. The values of the matrix are same as in square model shape matrix. Figure 2.22 shows an example, where n = 5 and m =12. Its polar model of shape matrix is

$$PSM = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

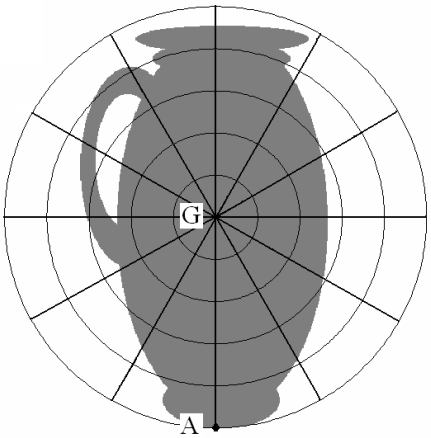


Figure 2.22: Polar model shape

Polar model of shape matrix is simpler than square model because it only uses one matrix no matter how many maximum radii are on the shape. However, since the sampling density is not constant with the polar sampling raster, a weighed shape matrix is necessary. For the detail, refer to [20].

The shape matrix exists for every compact shape. There is no limit to the scope of the shapes that the shape matrix can represent. It can describe even shapes with holes. Shape matrix is also invariant under translation, rotation and scaling of the object. The shape of the object can be reconstructed from the shape matrix; the accuracy is given by the size of the grid cells.

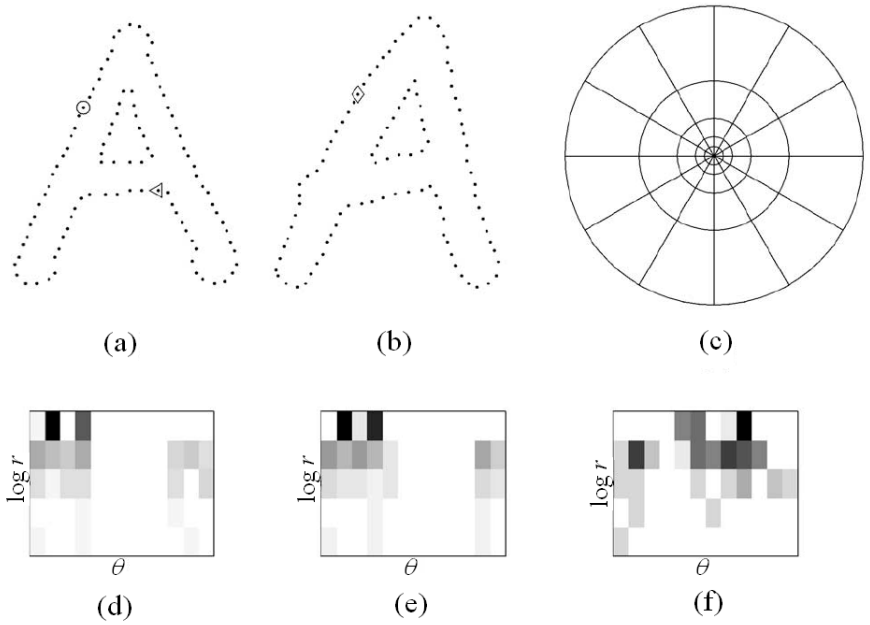
2.4.9 Shape context

In [22], the shape context has been shown to be a powerful tool for object recognition tasks. It is used to find corresponding features between model and image.

Shape contexts analysis begins by taking N samples from the edge elements on the shape. These points can be on internal or external contours. Consider the vectors originating from a point to all other sample points on the shape. These vectors express the appearance of the entire shape relative to the reference point. This descriptor is the histogram of the relative polar coordinates of all other points:

$$h_i(k) = \sharp\{Q \neq P_i \ : \ (Q - P_i) \in bin(k)\}$$

An example is shown in Figure 2.23. (c) is the diagram of log-polar histogram that has 5 bins for the polar direction and 12 bins for the angular direction. The histogram of a point P_i is formed by the following: putting the center of the histogram bins diagram on the point P_i, each bin of this histogram contains a count of all other sample points on the shape falling into that bin. Note on this figure, the shape contexts (histograms) for the points marked by $\circ$ (in (a)), $\diamond$ (in (b)) and $\triangleleft$ (in (a)) are shown in (d), (e) and (f), respectively. It is clear that the shape contexts for the points marked by $\circ$ and $\diamond$, which are computed for relatively similar points on the two shapes, have visual similarity. By contrast, the shape context for $\triangleleft$ is quite different from the others. Obviously, this descriptor is a rich description, since as N gets large, the representation of the shape becomes exact.



(a) and (b) Sampled edge points of two shapes; (c) diagram of log-polar histogram bins used in computing the shape contexts; (d), (e) and (f) shape contexts for reference sample points marked by $\circ$, $\diamond$ and $\triangleleft$ in (a) and (b), respectively. (Dark=large value).

Figure 2.23: Shape context computation and graph matching

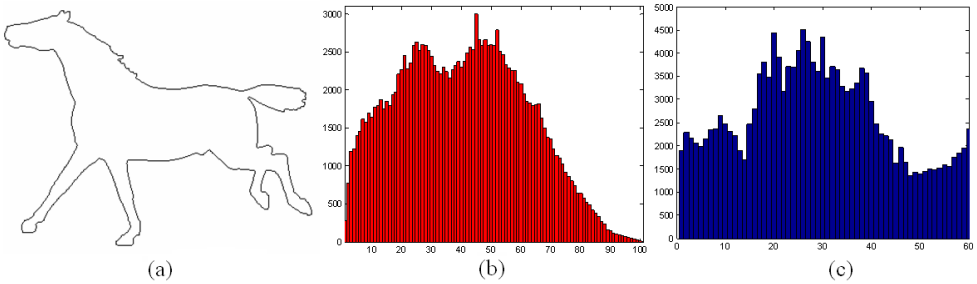
Shape context matching is often used to find the corresponding points on two shapes. It has been applied to a variety of object recognition problems [22, 62, 63, 64]. The shape context descriptor has the following invariance properties:

- translation: the shape context descriptor is inherently translation invariant as it is based on relative point locations.
- scaling: for clutter-free images the descriptor can be made scale invariant by normalizing the radial distances by the mean (or median) distance between all point pairs.
- rotation: it can be made rotation invariant by rotating the coordinate system at each point so that the positive x-axis is aligned with the tangent vector.

- shape variation: the shape context is robust against slight shape variations.
- few outliers: points with a final matching cost larger than a threshold value are classified as outliers. Additional 'dummy' points are introduced to decrease the effects of outliers.

2.4.10 Chord distribution

The basic idea of chord distribution is to calculate the lengths of all chords in the shape (all pair-wise distances between boundary points) and to build a histogram of their lengths and orientations [65]. The "lengths" histogram is invariant to rotation and scales linearly with the size of the object. The "angles" histogram is invariant to object size and shifts relative to object rotation. Figure 2.24 gives an example of chord distribution.



(a) Original contour; (b) chord lengths histogram; (c) chord angles histogram (each stem covers 3 degrees).

Figure 2.24: Chord distribution

2.4.11 Shock graphs

Shock graphs is a descriptor based on the medial axis. The medial axis is the most popular that has been proposed as a useful shape abstraction tool for the representation and modeling of animate shapes. Skeleton and medial axes have been extensively used for characterizing objects satisfactorily using structures that are composed of line or arc patterns. Medial axis is an image processing operation which reduces input shapes to axial stick-like representations. It is as the loci of centers of bi-tangent circles that fit entirely within the foreground region being considered. Figure 2.25 illustrates the medial axis for a rectangular shape.

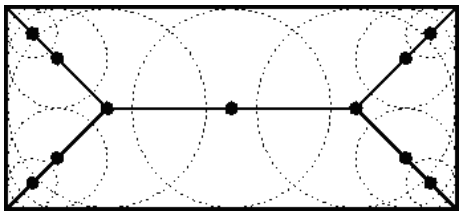


Figure 2.25: Medial axis of a rectangle defined in terms of bi-tangent circles

We notice that the radius of each circle is variable. This variable is a function of the loci of points on the medial axis. We call this function as the radius function.

A shock graph is a shape abstraction that decomposes a shape into a set of hierarchically organized primitive parts. Siddiqi and Kimia define the concept of a shock graph [66] as an

abstraction of the medial axis of a shape onto a directed acyclic graph (DAG). Shock segments are curve segments of the medial axis with monotonic flow, and give a more refined partition of the medial axis segments. Figure 2.26 is for example.

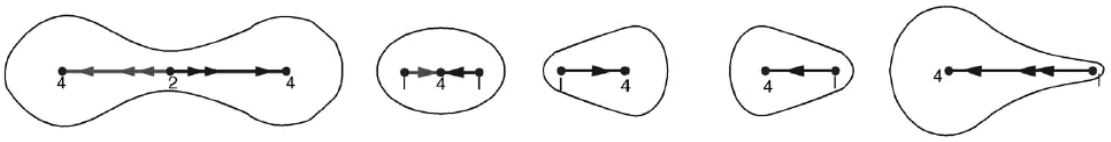


Figure 2.26: Shock segments

The skeleton points are first labeled according to the local variation of the radius function at each point. Shock graph can distinguish the shapes but the medial axis cannot. Figure 2.27 shows two examples of shapes and their shock graphs.

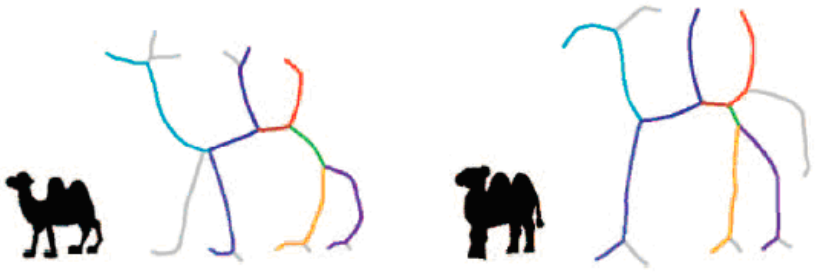


Figure 2.27: Examples of shapes and their shock graphs

To calculate the distance between two shock graphs, in [67], the authors employ a polynomial-time edit-distance algorithm. It shows this algorithm has the good performances for boundary perturbations, articulation and deformation of parts, segmentation errors, scale variations, viewpoint variations and partial occultation. However the authors also indicate the computation complexity is very high. The matching shape typically takes about 3-5 minutes on an SGI Indigo II (195 MHz), which limits the number of shapes that can be practically matched.

2.4.12 Discussions

Spacial feature descriptor is a direct method to describe a shape. These descriptors can apply the theory of tree-based (Adaptive grid resolution and Convex hull), statistic (Chain code histogram, Beam angle statistics, Shape context and Chord distribution) or syntactic analysis (Smooth curve decomposition) to extract or represent the feature of a shape. This description scheme not only compresses the data of a shape, but also provides a compact and meaningful form to facilitate further recognition operations.

2.5 Moments

The concept of moment in mathematics evolved from the concept of moment in physics. It is an integrated theory system. For both contour and region of a shape, one can use moment's theory to analyze the object.

2.5.1 Boundary moments

Boundary moments, analysis of a contour, can be used to reduce the dimension of boundary representation [55]. Assume shape boundary has been represented as a 1-D shape representation $z(i)$ as introduced in Section 2.2, the r^{th} moment m_r and central moment μ_r can be estimated as

$$m_r = \frac{1}{N}\sum_{i=1}^{N}[z(i)]^r \quad and \quad \mu_r = \frac{1}{N}\sum_{i=1}^{N}[z(i) - m_1]^r$$

where N is the number of boundary points.

The normalized moments $\bar{m_r} = m_r/(\mu_2)^{r/2}$and $\bar{\mu_r} = \mu_r/(\mu_2)^{r/2}$ are invariant to shape translation, rotation and scaling. Less noise-sensitive shape descriptors can be obtained from

$$F_1 = \frac{(\mu_2)^{1/2}}{m_1}, \quad F_2 = \frac{\mu_3}{(\mu_2)^{3/2}} \quad and \quad F_3 = \frac{\mu_4}{(\mu_2)^2},$$

The other boundary moments method treats the 1-D shape feature function $z(i)$ as a random variable $\mathbf{v}$ and creates a K bins histogram $p(v_i)$ from $z(i)$. Then, the r^{th} central moment is obtained by

$$\mu_r = \sum_{i=1}^{K}(v_i - m)^r\, p(v_i) \quad and \quad m = \sum_{i=1}^{K} v_i p(v_i)$$

The advantage of boundary moment descriptors is that it is easy to implement. However, it is difficult to associate higher order moments with physical interpretation.

2.5.2 Region moments

Among the region-based descriptors, moments are very popular. These include invariant moments, Zernike moments, Radial Chebyshev moments, etc.

The general form of a moment function m_{pq} of order $(p+q)$ of a shape region can be given as:

$$m_{pq} = \sum_x \sum_y \Psi_{pq}(x,y) f(x,y) \qquad p,q = 0,1,2\cdots$$

where Ψ_{pq} is known as the *moment weighting kernel* or the *basis set*; $f(x,y)$ is the shape region Eq. 1.1.

Invariant moments (IM)

Invariant moments (IM) are also called geometric moment invariants. Geometric moments, are the simplest of the moment functions with basis $\Psi_{pq} = x^p y^q$, while complete, is not orthogonal [57]. Geometric moment function m_{pq} of order $(p+q)$ is given as:

$$m_{pq} = \sum_x \sum_y x^p y^q f(x,y) \qquad p,q = 0,1,2\cdots$$

The geometric central moments, which are invariant to translation, are defined as:

$$\mu_{pq} = \sum_x \sum_y (x-\bar{x})^p\,(y-\bar{y})^q\, f(x,y) \quad p,q = 0,1,2\cdots$$

where $\bar{x} = m_{10}/m_{00}$ and $\bar{y} = m_{01}/m_{00}$

A set of 7 invariant moments (IM) are given by [57]:

$\phi_1 = \eta_{20} + \eta_{02}$

$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$

$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$

$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$

$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})$
$\cdot \left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]$

$\phi_6 = (\eta_{20} - \eta_{02})\left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] + 4\eta_{11}^2(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$

$\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] + (3\eta_{12} - \eta_{03})(\eta_{21} + \eta_{03})$
$\cdot \left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]$

where $\eta_{pq} = \mu_{pq}/\mu_{00}^{\gamma}$ and $\gamma = 1 + (p+q)/2$ for $p + q = 2, 3, \cdots$

IM are computationally simple. Moreover, they are invariant to rotation, scaling and translation. However, they have several drawbacks [10]:

- information redundancy: since the basis is not orthogonal, these moments suffer from a high degree of information redundancy.
- noise sensitivity: higher-order moments are very sensitive to noise.
- large variation in the dynamic range of values: since the basis involves powers of p and q, the moments computed have large variation in the dynamic range of values for different orders. This may cause numerical instability when the image size is large.

Algebraic moment invariants

The algebraic moment invariants are computed from the first m central moments and are given as the eigenvalues of predefined matrices, $M_{[j,k]}$, whose elements are scaled factors of the central moments [68]. The algebraic moment invariants can be constructed up to arbitrary order and are invariant to affine transformations. However, algebraic moment invariants performed either very well or very poorly on the objects with different configuration of outlines.

Zernike moments (ZM)

Zernike Moments (ZM) are orthogonal moments [10]. The complex Zernike moments are derived from orthogonal Zernike polynomials:

$$V_{nm}(x, y) = V_{nm}(r\cos\theta, \sin\theta) = R_{nm}(r)\exp(jm\theta)$$

where $R_{nm}(r)$is the orthogonal radial polynomial:

$$R_{nm}(r) = \sum_{s=0}^{(n-|m|)/2} (-1)^s \frac{(n-s)!}{s! \times \left(\frac{n-2s+|m|}{2}\right)! \left(\frac{n-2s-|m|}{2}\right)!} r^{n-2s}$$

$n = 0, 1, 2, \cdots$; $0 \leq |m| \leq n$; and $n - |m|$ is even.

Zernike polynomials are a complete set of complex valued functions orthogonal over the unit disk, i.e., $x^2 + y^2 \leq 1$. The Zernike moment of order n with repetition m of shape region $f(x, y)$ (Eq. 1.1) is given by:

$$Z_{nm} = \frac{n+1}{\pi} \sum_r \sum_\theta f(r\cos\theta, r\sin\theta) \cdot R_{nm}(r) \cdot exp(jm\theta) \quad r \leq 1$$

Zernike moments (ZM) have the following advantages [69]:

- rotation invariance: the magnitudes of Zernike moments are invariant to rotation.
- robustness: they are robust to noise and minor variations in shape.
- expressiveness: since the basis is orthogonal, they have minimum information redundancy.

However, the computation of ZM (in general, continuous orthogonal moments) pose several problems:

- coordinate space normalization: the image coordinate space must be transformed to the domain where the orthogonal polynomial is defined (unit circle for the Zernike polynomial).
- numerical approximation of continuous integrals: the continuous integrals must be approximated by discrete summations. This approximation not only leads to numerical errors in the computed moments, but also severely affects the analytical properties such as rotational invariance and orthogonality.
- computational complexity: computational complexity of the radial Zernike polynomial increases as the order becomes large.

Radial Chebyshev moments (RCM)

The radial Chebyshev moment of order p and repetition q is defined as [70]:

$$S_{pq} = \frac{1}{2\pi\rho(p,m)} \sum_{r=0}^{m-1} \sum_{\theta=0}^{2\pi} t_p(r) \cdot exp(-jq\theta) \cdot f(r,\theta)$$

where $t_p(r)$ is the scaled orthogonal Chebyshev polynomials for an image of size $N \times N$:

$$t_0(x) = 1$$
$$t_1(x) = (2x - N + 1)/N$$
$$t_p(x) = \frac{(2p-1)t_1(x)t_{p-1}(x) - (p-1)\left\{1 - \frac{(p-1)^2}{N^2}\right\} t_{p-2}(x)}{p}, \quad p > 1$$

$\rho(p, N)$ is the squared-norm:

$$\rho(p,N) = \frac{N\left(1-\frac{1}{N^2}\right)\left(1-\frac{2^2}{N^2}\right)\cdots\left(1-\frac{p^2}{N^2}\right)}{2p+1} \qquad p = 0, 1, \cdots, N-1$$

and $m = (N/2) + 1$.

The mapping between (r, θ) and image coordinates (x, y) is given by:

$$x = \frac{rN}{2(m-1)}\cos(\theta) + \frac{N}{2}$$
$$y = \frac{rN}{2(m-1)}\sin(\theta) + \frac{N}{2}$$

Compared to Chebyshev moments, radial Chebyshev moments possess rotational invariance property.

2.5.3 Discussions

Besides the previous moments, there are other moments for shape representation, for example, homocentric polar-radius moment [71], orthogonal Fourier-Mellin moments (OFMMs) [72], pseudo-Zernike Moments [73], etc. The study shows that the moment-based shape descriptors are usually concise, robust and easy to compute. It is also invariant to scaling, rotation and translation of the object. However, because of their global nature, the disadvantage of moment-based methods is that it is difficult to correlate high order moments with a shape's salient features.

2.6 Scale space approaches

In scale space theory a curve is embedded into a continuous family $\{\Gamma_\sigma : \sigma \geq 0\}$ of gradually simplified versions. The main idea of scale spaces is that the original curve $\Gamma = \Gamma_0$ should get more and more simplified, and so small structures should vanish as parameter σ increases. Thus due to different scales (values of σ), it is possible to separate small details from relevant shape properties. The ordered sequence $\{\Gamma_\sigma : \sigma \geq 0\}$ is referred to as evolution of Γ. Scale-spaces find wide application in computer vision, in particular, due to smoothing and elimination of small details.

A lot of shape features can be analyzed in scale-space theory to get more information about shapes. Here we introduced 2 scale-space approaches: curvature scale-space (CSS) and intersection points map (IPM).

2.6.1 Curvature scale-space

The curvature scale-space (CSS) method, proposed by F. Mokhtarian in 1988, was selected as a contour shape descriptor for MPEG-7. This approach is based on multi-scale representation and curvature to represent planar curves. For convenience, we copy the nature parametrization equation (Eq. 1.2) as following:

$$\Gamma(\mu) = (x(\mu), y(\mu)) \tag{2.4}$$

An evolved version of that curve is defined by

$$\Gamma_\sigma(\mu) = (X(\mu, \sigma), Y(\mu, \sigma))$$

where $X(\mu, \sigma) = x(\mu) * g(\mu, \sigma)$ and $Y(\mu, \sigma) = y(\mu) * g(\mu, \sigma)$, $*$ is the convolution operator, and $g(\mu, \sigma)$ denotes a Gaussian filter with standard deviation σ defined by

$$g(\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} exp(\frac{-\mu^2}{2\sigma^2})$$

Functions $X(\mu, \sigma)$ and $Y(\mu, \sigma)$ are given explicitly by

$$X(\mu, \sigma) = \int_{-\infty}^{\infty} x(v) \frac{1}{\sigma\sqrt{2\pi}} exp(\frac{-(\mu - v)^2}{2\sigma^2}) dv$$

$$Y(\mu, \sigma) = \int_{-\infty}^{\infty} y(v) \frac{1}{\sigma\sqrt{2\pi}} exp(\frac{-(\mu - v)^2}{2\sigma^2}) dv$$

The curvature of is given by

$$k(\mu,\sigma) = \frac{X_\mu(\mu,\sigma)Y_{\mu\mu}(\mu,\sigma) - X_{\mu\mu}(\mu,\sigma)Y_\mu(\mu,\sigma)}{(X_\mu(\mu,\sigma)^2 - Y_\mu(\mu,\sigma)^2)^{3/2}}$$

where

$$X_\mu(\mu,\sigma) = \frac{\partial}{\partial\mu}(x(\mu) * g(\mu,\sigma)) = x(\mu) * g_\mu(\mu,\sigma)$$

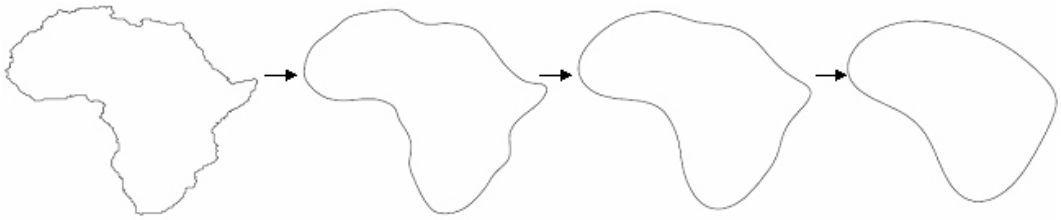
$$X_{\mu\mu}(\mu,\sigma) = \frac{\partial^2}{\partial\mu^2}(x(\mu) * g(\mu,\sigma)) = x(\mu) * g_{\mu\mu}(\mu,\sigma)$$

$$Y_\mu(\mu,\sigma) = \frac{\partial}{\partial\mu}(y(\mu) * g(\mu,\sigma)) = y(\mu) * g_\mu(\mu,\sigma)$$

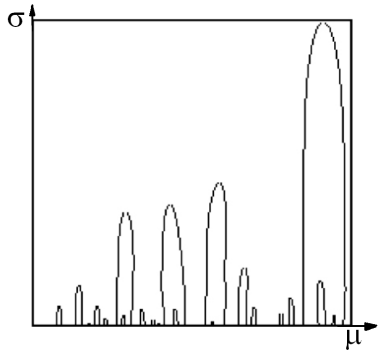
$$Y_{\mu\mu}(\mu,\sigma) = \frac{\partial^2}{\partial\mu^2}(y(\mu) * g(\mu,\sigma)) = y(\mu) * g_{\mu\mu}(\mu,\sigma)$$

Note that σ is also referred to as a scale parameter. The process of generating evolved versions of Γ_σ as σ increases from 0 to ∞ is referred to as the evolution of Γ_σ. This technique is suitable for removing noise and smoothing a planar curve as well as gradual simplification of a shape.

The function defined by $k(\mu,\sigma) = 0$ is the CSS image of Γ. Figure 2.28 is a CSS image examples.



(a)



(b)

(a) Evolution of Africa: from left to right $\sigma = 0$(original), $\sigma = 4$, $\sigma = 8$ and $\sigma = 16$, respectively; (b) CSS image of Africa.

Figure 2.28: Curvature scale-space image

The representation of CSS is the maxima of CSS contour of an image. Many methods for representing the maxima of CSS exist in the literatures [74, 75, 46] and the CSS technique has been shown to be robust contour-based shape representation technique. The basic properties of the CSS representation are as follows:

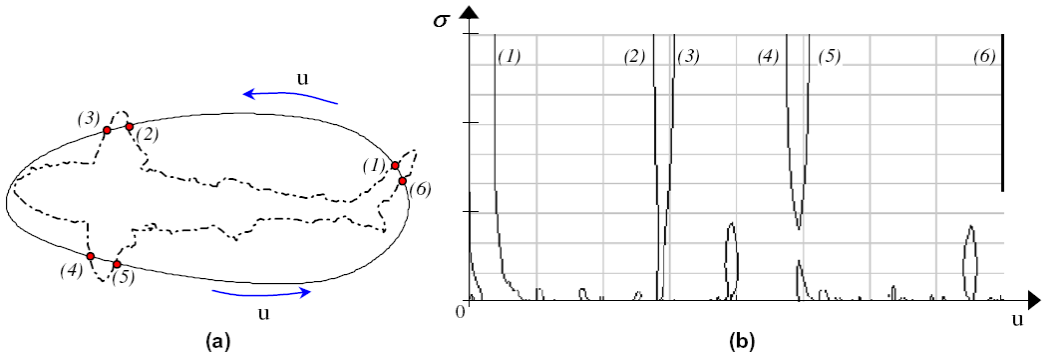
- it captures the main features of a shape, enabling similarity-based retrieval;

- it is robust to noise, changes in scale and orientation of objects;
- it is compact, reliable and fast;
- It retains the local information of a shape. Every concavity or convexity on the shape has its own corresponding contour on the CSS image.

Although CSS has a lot of advantages, it does not always give results in accordance with human vision system. The main drawbacks of this description are due to the problem of shallow concavities/convexities on a shape. It can be shown that the shallow and deep concavities/convexities may create the same large contours on the CSS image. In [76, 77], the authors gave some methods to alleviate these effects.

2.6.2 Intersection points map

Similarly to the CSS, many methods also use a Gaussian kernel to progressively smooth the curve relatively to the varying bandwidth. In [78], the authors proposed a new algorithm, intersection points map (IPM), based on this principle, instead of characterizing the curve with its curvature involving 2^{nd} order derivatives, it uses the intersection points between the smoothed curve and the original. As the standard deviation of the Gaussian kernel increases, the number of the intersection points decreases. By analyzing these remaining points, features for a pattern can be defined. Since this method deals only with curve smoothing, it needs only the convolution operation in the smoothing process. So this method is faster than the CSS one with equivalent performances. Figure 2.29 is an example of IPM.



(a) An original contour; (b) an IPM image in the (u, σ) plane. The IPM points indicated by (1)–(6) refer to the corresponding intersection points in (a).

Figure 2.29: Example of the IPM

The IPM pattern can be identified regardless of its orientation, translation and scale change. It is also resistant to noise for a range of noise energy. The main weakness of this approach is that it fails to handle occulted contours and those having undergone a non-rigid deformation.

2.6.3 Discussions

As multi-resolution analysis in signal processing, scale-space theory can obtain abundant information about a contour with different scales. In scale-space, global pattern information can be interpreted from higher scales, while detailed pattern information can be interpreted from

lower scales. Scale-space algorithm benefit from the boundary information redundancy in the new image, making it less sensitive to errors in the alignment or contour extraction algorithms. The great advantages are the high robustness to noise and the great coherence with human perception.

2.7 Shape transform domains

The transform domain class includes methods which are formed by the transform of the detected object or the transform of the whole image. Transforms can therefore be used to characterize the appearance of images. The shape feature is represented by the all or partial coefficients of a transform.

2.7.1 Fourier descriptors

Although, Fourier descriptor (FD) is a 40-year-old technique, it is still considered as a valid description tool. The shape description and classification using FD either in contours or regions are simple to compute, robust to noise and compact. It has many applications in different areas.

One-dimensional Fourier descriptors

In general, Fourier descriptor (FD) is obtained by applying Fourier transform on a shape signature that is a one-dimensional function which is derived from shape boundary coordinates (cf. Section 2.2). The normalized Fourier transformed coefficients are called the Fourier descriptor of the shape. FD derived from different signatures has significant different performance on shape retrieval. As shown in [38, 75], FD derived from centroid distance function *r(t)* outperforms FD derived from other shape signatures in terms of overall performance. The discrete Fourier transform of *r(t)* is then given by

$$a_n = \frac{1}{N}\sum_{t=0}^{N-1} r(t) exp\left(\frac{-j2\pi nt}{N}\right), \quad n = 0, 1, \cdots, N-1$$

Since the centroid distance function *r(t)* is only invariant to rotation and translation, the acquired Fourier coefficients have to be further normalized so that they are scaling and start point independent shape descriptors. From Fourier transform theory, the general form of the Fourier coefficients of a contour centroid distance function $r(t)$ transformed through scaling and change of start point from the original function $r(t)^{(o)}$ is given by

$$a_n = exp(jn\tau) \cdot s \cdot a_n^{(o)}$$

where a_n and $a_n^{(o)}$ are the Fourier coefficients of the transformed shape and the original shape, respectively, τ is the angles incurred by the change of start point; s is the scale factor. Now considering the following expression:

$$b_n = \frac{a_n}{a_1} = \frac{exp(jn\tau) \cdot s \cdot a_n^{(o)}}{exp(j\tau) \cdot s \cdot a_1^{(o)}} = \frac{a_n^{(o)}}{a_1^{(o)}} exp[j(n-1)\tau] = b_n^{(o)} exp[j(n-1)\tau]$$

where b_n and $b_n^{(o)}$ are the normalized Fourier coefficients of the transformed shape and the original shape, respectively. If we ignore the phase information and only use magnitude of the coefficients,

then $|b_n|$ and $\left|b_n^{(o)}\right|$ are the same. In other words, $|b_n|$ is invariant to translation, rotation, scaling and change of start point.

The set of magnitudes of the normalized Fourier coefficients of the shape $\{|b_n|,\ 0 < n < N\}$ are used as shape descriptors, denoted as

$$\{FD_n,\ 0 < n < N\}.$$

One-dimensional FD has several nice characteristics such as simple derivation, simple normalization and simple to do matching. As indicated by [75], for efficient retrieval, 10 FDs are sufficient for shape description.

Region-based Fourier descriptor

The region-based FD is referred to as generic FD (GFD), which can be used for general applications. Basically, GFD is derived by applying a modified polar Fourier transform (MPFT) on shape image [79, 39]. In order to apply MPFT, the polar shape image is treated as a normal rectangular image. The steps are

1. the approximated normalized image is rotated counter clockwise by an angular step sufficiently small.
2. the pixel values along positive x-direction starting from the image center are copied and pasted into a new matrix as row elements.
3. the steps 1 and 2 are repeated until the image is rotated by 360°.

The result of these steps is that an image in polar space plots into Cartesian space.

Figure 2.30 shows the polar shape image turning into normal rectangular image.

(a) (b)

(a) Original shape image in polar space; (b) polar image of (a) plotted into Cartesian space.

Figure 2.30: The polar shape image turns into normal rectangular image.

The Fourier transform is acquired by applying a discrete 2D Fourier transform on this shape image.

$$pf(\rho,\phi) = \sum_r \sum_i f(r,\theta_i) exp[j2\pi(\frac{r}{R}\rho + \frac{2\pi i}{T}\phi)]$$

where $0 \leq r = \sqrt{[(x-g_x)^2 + (y-g_y)^2]} < R$, and $\theta_i = i(2\pi/T)$; $0 \leq \rho < R$, $0 \leq \phi < T$. (g_x, g_y) is the center of mass of the shape; R and T are the radial and angular resolutions. The acquired Fourier coefficients are translation invariant. Rotation and scaling invariance are achieved by the following:

$$GFD = \left\{ \frac{|pf(0,0)|}{area}, \frac{|pf(0,1)|}{|pf(0,0)|}, \cdots, \frac{|pf(0,n)|}{|pf(0,0)|}, \cdots, \frac{|pf(m,0)|}{|pf(0,0)|}, \cdots, \frac{|pf(m,n)|}{|pf(0,0)|} \right\}$$

where *area* is the area of the bounding circle in which the polar image resides. m is the maximum number of the radial frequencies selected and n is the maximum number of angular frequencies selected. m and n can be adjusted to achieve hierarchical coarse to fine representation requirement.

For efficient shape description, in the implementation of [79], 36 GFD features reflecting $m = 4$ and $n = 9$ are selected to index the shape. The experimental results have shown GFD as invariant to translation, rotation, and scaling. For obtaining the affine and general minor distortions invariance, in [79], the authors proposed Enhanced Generic Fourier Descriptor (EGFD) to improve the GFD properties.

2.7.2 Wavelet transform

A hierarchical planar curve descriptor is developed by using the wavelet transform [80]. This descriptor decomposes a curve into components of different scales so that the coarsest scale components carry the global approximation information while the finer scale components contain the local detailed information. The wavelet descriptor has many desirable properties such as multi-resolution representation, invariance, uniqueness, stability, and spatial localization. In [81], the authors use dyadic wavelet transform deriving an affine invariant function. In [82], a descriptor is obtained by applying the Fourier transform along the axis of polar angle and the wavelet transform along the axis of radius. This feature is also invariant to translation, rotation, and scaling. At same time, the matching process of wavelet descriptor can be accomplished cheaply.

2.7.3 Angular radial transformation

The angular radial transformation (ART) is based in a polar coordinate system where the sinusoidal basis functions are defined on a unit disc. Given an image function in polar coordinates, $f(\rho, \theta)$, an ART coefficient F_{nm} (radial order n, angular order m) can be defined as [83]:

$$F_{nm} = \int_0^{2\pi} \int_0^1 V_{nm}(\rho, \theta) f(\rho, \theta) \rho d\rho d\theta$$

$V_{nm}(\rho, \theta)$ is the ART basis function and is separable in the angular and radial directions so that:

$$V_{nm}(\rho, \theta) = A_m(\theta) R_n(\rho)$$

The angular basis function, A_m, is an exponential function used to obtain orientation invariance. This function is defined as:

$$A_m(\theta) = \frac{1}{2\pi} e^{jm\theta}$$

R_n, the radial basis function, is defined as:

$$R_n(\rho) = \begin{cases} 1 & if\ n = 0 \\ 2\cos(\pi n \rho) & if\ n \neq 0 \end{cases}$$

In MPEG-7, twelve angular and three radial functions are used ($n < 3, m < 12$). Real parts of the 2-D basis functions are shown in Figure 2.31.

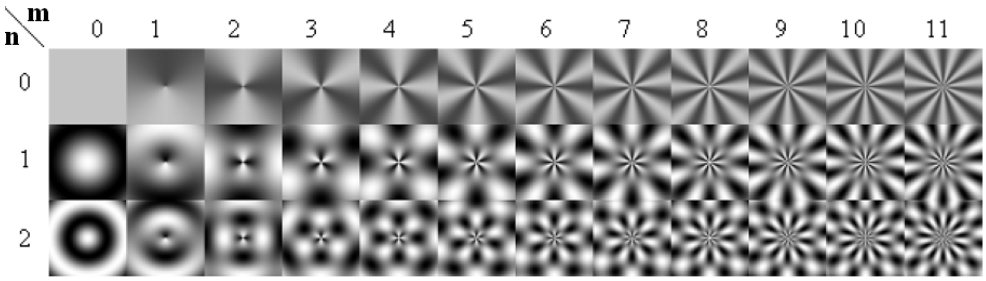


Figure 2.31: Real parts of the ART basis functions

For scale normalization, the ART coefficients are divided by the magnitude of ART coefficient of order $n = 0, m = 0$.

MPEG-7 standardization process showed the efficiency of 2-D angular radial transformation. This descriptor is robust to translations, scaling, multi-representation (remeshing, weak distortions) and noises.

2.7.4 Shape signature harmonic embedding

A harmonic function is obtained by a convolution between the Poisson kernel $P_R(r, \theta)$ and a given boundary function $u(Re^{j\phi})$. Poisson kernel is defined by

$$P_R(r, \theta) = \frac{R^2 - r^2}{R^2 - 2Rr\cos(\theta) + r^2}$$

The boundary function could be any real- or complex-valued function, but here we choose shape signature functions for the purpose of shape representation. For any shape signature $s[n], n = 0, 1, \cdots, N-1$, the boundary values for a unit disk can be set as

$$u(Re^{j\phi}) = u(Re^{j\omega_0 n}) = s[n]$$

where $\omega_0 = 2\pi/N$, $\phi = \omega_0 n$.

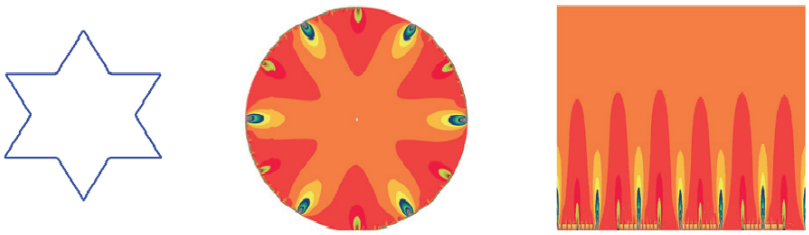
So the harmonic function u can be written as

$$u(re^{j\theta}) = \frac{1}{2\pi}\int_0^{2\pi} u(Re^{j\phi}) P_R(r, \phi - \theta) d\phi \tag{2.5}$$

The Poisson kernel $P_R(r, \theta)$ has a low-pass filter characteristic, where the radius r is inversely related to the bandwidth of the filter. The radius r is considered as scale parameter of a multi-scale representation [84]. Another important property is $P_R(0, \theta) = 1$, indicating $u(0)$ is the mean value of boundary function $u(Re^{j\phi})$.

In [84], the authors proposed a formulation of a discrete closed-form solution for the Poisson's integral formula Eq. 2.5, so that one can avoid the need for approximation or numerical calculation of the Poisson summation form.

As in Subsection 2.7.1, the harmonic function inside the disk can be mapped to a rectilinear space for a better illustration. Figure 2.32 shows an example for a star shape. Here, we used curvature as the signature to provide boundary values.

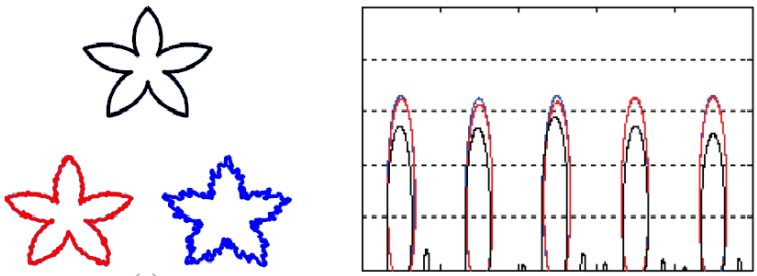


(a) (b) (c)

(a) Example shape; (b) harmonic function within the unit disk; (c) rectilinear mapping of the function.

Figure 2.32: Harmonic embedding of curvature signature

The zero-crossing image of the harmonic functions is extracted as shape feature. This shape descriptor is invariant to translation, rotation and scaling. It is also robust to noise. Figure 2.33 is for example. The original curve is corrupted with different noise levels, and the harmonic embeddings show robustness to the noise.



(a) (b)

(a) Original and noisy shapes; (b) harmonic embedding images for centroid distance signature.

Figure 2.33: centroid distance signature harmonic embedding that is robust to noisy boundaries

At same time, it is more efficient than CSS descriptor. However, it is not suitable for similarity retrieval, because it is unconsistent to non-rigid transform.

2.7.5 ℜ-Transform

The ℜ-Transform to represent a shape is based on the Radon transform. The approach is presented as follow. We assume that the function f is the domain of a shape, cf. Eq. 1.1. Its Radon transform is defined by:

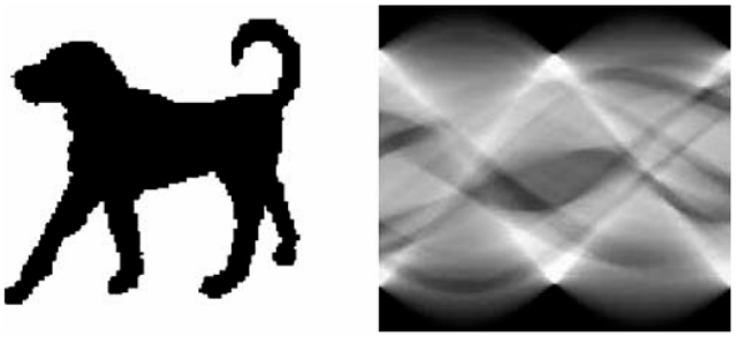
$$T_R(\rho,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(x\cos\theta + y\sin\theta - \rho)dxdy$$

where $\delta(.)$ is the Dirac delta-function:

$$\delta(x) = \begin{cases} 1 & if\ x = 0 \\ 0 & otherwise \end{cases}$$

$\theta \in [0, \pi]$ and $\rho \in (-\infty, \infty)$. In other words, Radon transform $T_R(\rho, \theta)$ is the integral of f over the line $L_{(\rho,\theta)}$ defined by $\rho = x\cos\theta + y\sin\theta$.

Figure 2.34 is an example of a shape and its Radon transform.



(a) Shape (b) Radon transform of (a)

Figure 2.34: A shape and its Radon transform

The following transform is defined as $\Re$-transform:

$$\Re_f(\theta) = \int_{-\infty}^{\infty} T_R^2(\rho, \theta) d\rho$$

where $T_R(\rho, \theta)$ is the Radon transform of the domain function f. In [85], the authors show the following properties of $\Re_f(\theta)$:

- periodicity: $\Re_f(\theta \pm \pi) = \Re_f(\theta)$
- rotation: a rotation of the image by an angle θ_0 implies a translation of the $\Re$-transform of θ_0: $\Re_f(\theta + \theta_0)$.
- translation: the $\Re$-transform is invariant under a translation of the shape f by a vector $\overrightarrow{u} = (x_0, y_0)$.
- scaling: a change of the scaling of the shape f induces a scaling in the amplitude only of the $\Re$-transform.

Given a large collection of shapes, one $\Re$-transform per shape is not efficient to distinguish from the others because the $\Re$-transform provides a highly compact shape representation. In this perspective, to improve the description, each shape is projected in the Radon space for different segmentation levels of the Chamfer distance transform. Chamfer distance transform is introduced in [86, 87] (See Appendix A for detail).

Given the distance transform of a shape, the distance image is segmented into N equidistant levels to keep the segmentation isotropic. For each distance level, pixels having a distance value superior to that level are selected and at each level of segmentation, an $\Re$-transform is computed. In this manner, both the internal structure and the boundaries of the shape are captured.

Since a rotation of the shape implies a corresponding shift of the $\Re$-transform. Therefore, a one-dimensional Fourier transform is applied on this function to obtain the rotation invariance. After the discrete one-dimensional Fourier transform F, $\Re$-transform descriptor vector is defined as follows:

$$RTD = (\frac{F\Re^1(\frac{\pi}{M})}{F\Re^1(0)}, \cdots, \frac{F\Re^1(\frac{i\pi}{M})}{F\Re^1(0)}, \cdots, \frac{F\Re^1(\pi)}{F\Re^1(0)}, \cdots, \frac{F\Re^N(\frac{\pi}{M})}{F\Re^N(0)}, \cdots, \frac{F\Re^N(\frac{i\pi}{M})}{F\Re^N(0)}, \cdots, \frac{F\Re^N(\pi)}{F\Re^N(0)})$$

where $i \in [1, M]$, M is the angular resolution. $F\Re^\alpha$ is the magnitude of Fourier transform to $\Re$-transform. $\alpha \in [1, N]$, is the segmentation level of Chamfer distance transform.

2.7.6 Shapelets descriptor

Shapelets descriptor was proposed to present a model for animate shapes and extracting meaningful parts of objects. The model assumes that animate shapes (2D simple closed curves) are formed by a linear superposition of a number of shape bases. A basis function $\psi(s; \mu, \sigma)$ is defined in [88]: $\mu \in [0, 1]$ indicates the location of the basis function relative to the domain of the observed curve, and σ is the scale of the function ψ. Figure 2.35 shows the shape of the basis function ψ at different σ values. It displays variety with different parameter and transforms.

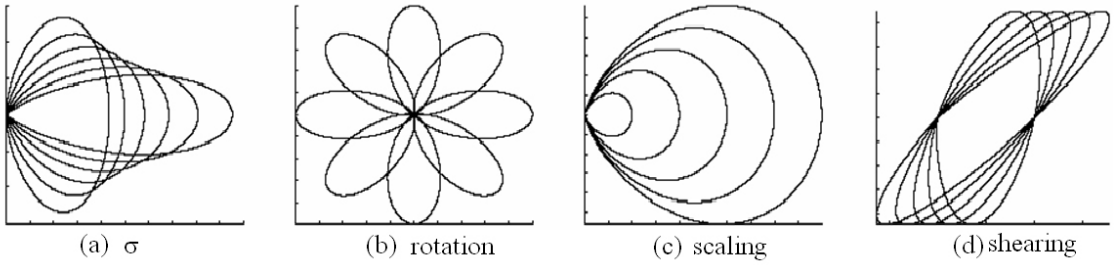


Figure 2.35: Each shape base is a lobe-shaped curve

The basis functions are subject to affine transformations by a 2×2 matrix of basis coefficients:

$$A_k = \begin{bmatrix} a_k & b_k \\ c_k & d_k \end{bmatrix}$$

The variables for describing a base are denoted by $\mathbf{b}_k = (A_k, \mu_k, \sigma_k)$ and are termed basis elements. The shapelet is defined by

$$\gamma(s; \mathbf{b}_k) = A_k \psi(s; \mu_k, \sigma_k)$$

Figure 2.35 (b,c,d) demonstrates shapelets obtained from the basis functions ψ by the affine transformations of rotation, scaling, and shearing respectively, as indicated by the basis coefficient A_k. By collecting all the shapelets at various μ, σ, A and discretizing them at multiple levels, an over-complete dictionary is obtained

$$\Delta = \{\gamma(s; \mathbf{b}_k) : \forall \mathbf{b};\ a\gamma_0,\ a > 0\}.$$

A special shapelet γ_0 is defined as an ellipse.

Shapelets are the building blocks for shape contours, and they form closed curves by linear addition:

$$\Gamma(s) = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \sum_{k=1}^{K} \begin{bmatrix} a_k & b_k \\ c_k & d_k \end{bmatrix} \psi(s; \mu_k, \sigma_k) + \mathbf{n}(s)$$

Here (x_0, y_0) is the centroid of the contour and $\mathbf{n}$ is residue.

A discrete representation $\mathbf{B} = (K, \mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_K)$, shown by the dots in second row of Figure 2.36, represents a shape. $\mathbf{B}$ is called the "shape script" by analogy to music scripts, where each shapelet is represented by a dot in the (μ, σ) domain. The horizontal axis is $\mu \in [0, 1]$ and the vertical axis is the σ. Large dots correspond to big coefficient matrix

$$A_k^2 = a_k^2 + b_k^2 + c_k^2 + d_k^2$$

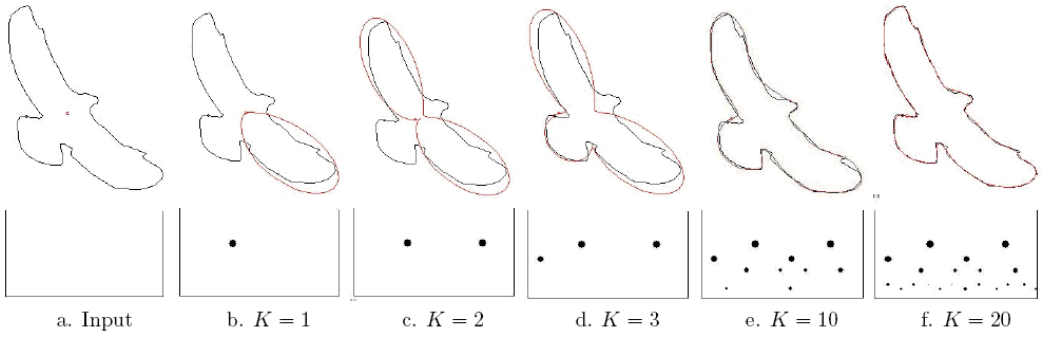


Figure 2.36: Pursuit of shape bases for an eagle contour

Clearly, computing the shape script $\mathbf{B}$ is a non-trivial task, since Δ is over-complete and there will be multiple sets of bases that reconstruct the curve with equal precision. [88] gave some pursuit algorithms to use shapelets representing a shape.

2.7.7 Discussions

As a kind of global shape description technique, shape analysis in transform domains takes the whole shape as the shape representation. The description scheme is designed for this representation. Unlike the spacial interrelation feature analysis, shape transform projects a shape contour or region into an other domain to obtain some of its intrinsic features. For shape description, there is always a trade-off between accuracy and efficiency. On one hand, shape should be described as accurate as possible; on the other hand, shape description should be as compact as possible to simplify indexing and retrieval. For a shape transform analysis algorithm, it is very flexible to accomplish a shape description with different accuracy and efficiency by choosing the number of transform coefficients.

2.8 Summary table

For convenience to compare these shape feature extraction approaches in this chapter, we summarize their properties in Table 2.1.

Shape representation			Recon-struc-ture	Invariance				Resistance			Computational complexity
				Translation	Rotation	Scale	Affine transform	Noise	Occultation	Non-rigid deformation	
Shape signatures	Complex coordinates		Yes	Bad	Bad	Bad	Bad	Average	Good	Bad	Low
	Central distance		No	Good	Good	Good	Bad	Average	Good	Bad	Low
	Tangent angle		No	Good	Good	Good	Bad	Bad	Good	Average	Low
	Curvature function		No	Good	Good	Good	Bad	Bad	Good	Average	Low
	Area function		No	Good	Good	Good	Good	Good	Good	Bad	Low
	Triangle-area representation		No	Good	Good	Good	Good	Good	Average	Average	Low
	Chord length function		No	Good	Good	Good	Bad	Bad	Bad	Bad	Low
Polygonal approximation	Merging	Distance threshold	No	Good	Good	Good	Bad	Good	Bad	Bad	Average
		Tunneling	No	Good	Good	Good	Bad	Good	Bad	Bad	Average
		Polygon evolution	No	Good	Good	Good	Bad	Good	Bad	Bad	Average
	Splitting		No	Good	Good	Good	Bad	Good	Bad	Bad	Average
Space interrelation Feature	Adaptive grid resolution		Yes	Good	Good	Good	Bad	Good	Good	Bad	Low
	Bounding box		Yes	Good	Good	Good	Average	Good	Good	Average	Average
	Convex hull		No	Good	Good	Good	Good	Average	Bad	Bad	High
	Chain code	Basic chain code	Yes	Good	Bad	Bad	Bad	Bad	Good	Bad	Low
		Vertex chain code	Yes	Good	Bad	Bad	Bad	Bad	Good	Bad	Low
		Statistic chain code	No	Good	Bad	Bad	Bad	Bad	Bad	Bad	Low
	Smooth curve decomposition		No	Good	Good	Good	Bad	Good	Good	Average	Average
	ALI-based representation		No	Good	Good	Good	Average	Good	Average	Bad	Average
	Beam angle statistics		No	Good	Good	Good	Bad	Good	Bad	Bad	Low
	Shape matrix	Square model	Yes	Good	Good	Good	Bad	Bad	Good	Bad	Average
		Polar model	Yes	Good	Good	Good	Bad	Bad	Good	Bad	Low
	Shape context		No	Good	Good	Good	Bad	Bad	Average	Average	Average
	Chord distribution		No	Good	Good	Good	Bad	Good	Bad	Bad	Low
	Shock graphs		Yes	Good	Good	Good	Good	Good	Good	Good	High
Moments	Boundary moments		No	Good	Good	Good	Bad	Average	Bad	Bad	Low
	Region moments	Invariant moments	No	Good	Good	Good	Bad	Bad	Bad	Bad	Average
		Algebraic Moment	No	Good	Good	Good	Good	Average	Bad	Bad	Average
		Zernike Moments	No	Good	Good	Good	Bad	Good	Average	Average	High
		Radial Chebyshev Moments	No	Good	Good	Good	Bad	Good	Average	Average	High
Scale-space methods	Curvature scale space		No	Good	Good	Good	Average	Good	Good	Average	Average
	Intersection points map		No	Good	Good	Good	Average	Good	Good	Bad	Average
Shape transform domains	Fourier descriptors	1-D Fourier descriptor	No	Good	Good	Good	Bad	Bad	Bad	Bad	Average
		Region-based Fourier descriptor	No	Good	Good	Good	Good	Good	Average	Average	High
	Wavelet transform		No	Good	Good	Good	Good	Average	Average	Bad	Average
	Angular radial transformation		No	Good	Good	Good	Bad	Good	Bad	Bad	High
	Signature harmonic embedding		No	Good	Good	Good	Average	Good	Average	Bad	High
	$\Re$ -Transform		No	Good	Good	Good	Bad	Good	Average	Average	High
	Shapelets descriptor		No	Good	Good	Good	Bad	Good	Bad	Bad	High

Table 2.1: Properties of shape feature extraction approaches

Frankly speaking, it is not equitable to affirm a property of an approach by rudely speaking "good" or "bad". Because certain approaches have great different performances under different conditions. For example, the method area function is invariant with affine transform under the condition of the contours sampled at its same vertices; whereas it is not robust to affine transform if the condition can't be contented. In addition, some approaches have good properties for certain type shapes; however it is not for the others. For example, the method shapelets representation is especially suitable for blobby objects, and it has shortcomings in representing elongated objects. So the simple evaluations in this table are only as a reference. These evaluations are drawn by assuming that all the necessary conditions have been contented for each approach.

2.9 Conclusion

In this chapter we made a study and a comparison the methods of shape-based feature extraction and representation. About 40 techniques for extraction of shape features have been shortly described and compared. Unlike the traditional classification, the approaches of shape-based feature extraction and representation were classified by their processing approaches. These processing approaches included shape signatures, polygonal approximation methods, spatial interrelation feature, moments approaches, scale-space methods and shape transform domains: in such way, one can easily select the appropriate processing approach. A synthetic table has been established for a fast and global comparison of the performances of these approaches.

Extracting a shape feature in accordance with human perception is not an easy task. Due to the fact that human vision and perception are an extraordinary complicated system, it is a utopia to hope that the machine vision has super excellent performance with small complexity. In addition, choosing appropriate features for a shape recognition system must consider what kinds of features are suitable for the task. There exists no general feature which would work best for every kind of images.

In the next chapters, we will propose three approaches about shape-based feature extractions and representations. The first two approaches proposed in Chapter 3, are used for the purpose of affine invariance; the last approach proposed in Chapter 4, is aimed at similarity-based measure.

Part II

Proposed approaches

Chapter 3

Affine invariant shape description: two approaches

3.1 Introduction

Shape distortion, arising from observing an object under arbitrary orientations, can be most appropriately described as a perspective transformation [54]. However, when the dimensions of the object are small compared to the distance from the camera to the object, a weak perspective can be assumed. In this case, the orthographic projection may be used as an approximation to the perspective projection, and the perspective distortion of the object can be modeled by shearing in the image plane. Furthermore, the affine transformation, which consists of rotation, translation, scaling, and shearing (skewing) transformations, may be used as an approximation to the perspective transformation [89].

Numerous shape representations have been suggested to recognize shapes under affine transforms. The most commonly used ones are affine arc length [90], enclosed area [91], affine invariant Fourier descriptors [89], moment invariants [92], curvature scale space [93, 94], linear signal space decomposition [95], triangular kernel [96] and dyadic wavelet transform [16, 91, 81]. In these methods, the basic idea is to use a parametrization which is robust with respect to affine transforms.

In this chapter, to extract an affine invariant attribute, two new algorithms are proposed based on equal area contour normalization. We analyze the relation of original contour and its filtered contour, by choosing the appropriate filter scale parameter, and then we extract affine invariant features. Experimental results indicate that the proposed methods are unaffected by boundary starting point variations and affine transforms even in the case of high deformations and serious noise contamination on the shapes.

The rest of the chapter is organized into four main parts. First, the background of affine transformation is introduced in Section 3.2. Second, we propose a method of equal area contour normalization in Section 3.3. Third, an approach of scale-controlled area difference shape descriptor is proposed and studied in Section 3.4. Fourth, the other affine invariant descriptor, the enclosed area affine invariant function, is proposed and its implementation details are studied in Section 3.5. For each method, a conclusion is locally given and finally, the comparison of the two proposed methods is discussed and the content of the chapter is summarized in Section 3.6.

3.2 Background of affine transformation

Consider a parametric closed curve $\Gamma(\mu) = (x(\mu), y(\mu))$ with parameter μ in a plane. A point on the curve under an affine transformation becomes

$$\begin{cases} x_a(\mu) = ax(\mu) + by(\mu) + e \\ y_a(\mu) = cx(\mu) + dy(\mu) + f \end{cases} \tag{3.1}$$

Eq. 3.1 can be rewritten in matrix form as follows:

$$\begin{bmatrix} x_a(\mu) \\ y_a(\mu) \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x(\mu) \\ y(\mu) \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} = A \begin{bmatrix} x(\mu) \\ y(\mu) \end{bmatrix} + B \tag{3.2}$$

where $x_a(\mu)$ and $y_a(\mu)$ represent the coordinates of the transformed shape. Translation is represented by matrix B, while scaling, rotation and shear are reflected in matrix A. The corresponding values of coefficients of A respective to different transformations can be found in the following matrices:

$$A_{Scaling} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix},\ A_{Rotation} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix},\ A_{Shear} = \begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix}$$

If S_x is equal to S_y, $A_{Scaling}$ represents uniform scaling, i.e., the length and width of the contour change with same proportion. θ is the rotation angle of the contour. k is the parameter of skewing degree. A shape is not deformed under rotation, uniform scaling and translation. However, non-uniform scaling and shear contribute to shape deformation under general affine transforms.

As introduced in Chapter 1, the arc length parameter observed on a closed contour transforms linearly under any linear transformation up to the similarity transform. Translation and rotation do not affect the arc length; scaling changes the parameter by the same amount. An arbitrary choice of a starting point only introduces a shift in the parameter. However, the arc length is nonlinearly transformed under an affine transform and would not be a suitable parameter in this situation [91].

There are two parameters which are linear under affine transforms. They are the affine arc length and the enclosed area.

The first parameter can be derived from the properties of determinants. It is defined as follows:

$$\tau = \int_{\alpha}^{\beta} \left[x'(s)y''(s) - x''(s)y'(s)\right]^{1/3} ds \tag{3.3}$$

where $x(s)$ and $y(s)$ are the coordinates of points on the contour and α and β are the curvilinear abscissa of two points on it.

The second affine invariant parameter is the enclosed area, which is based on the property of affine transforms: under affine mapping, all areas are changed in the same ratio. Based on this property, Arbter et al. [89] defined a parameter ψ, which is linear under a general affine transform, as follows:

$$\psi = \frac{1}{2} \int_{\alpha}^{\beta} \left|x(s)y'(s) - x'(s)y(s)\right| ds \tag{3.4}$$

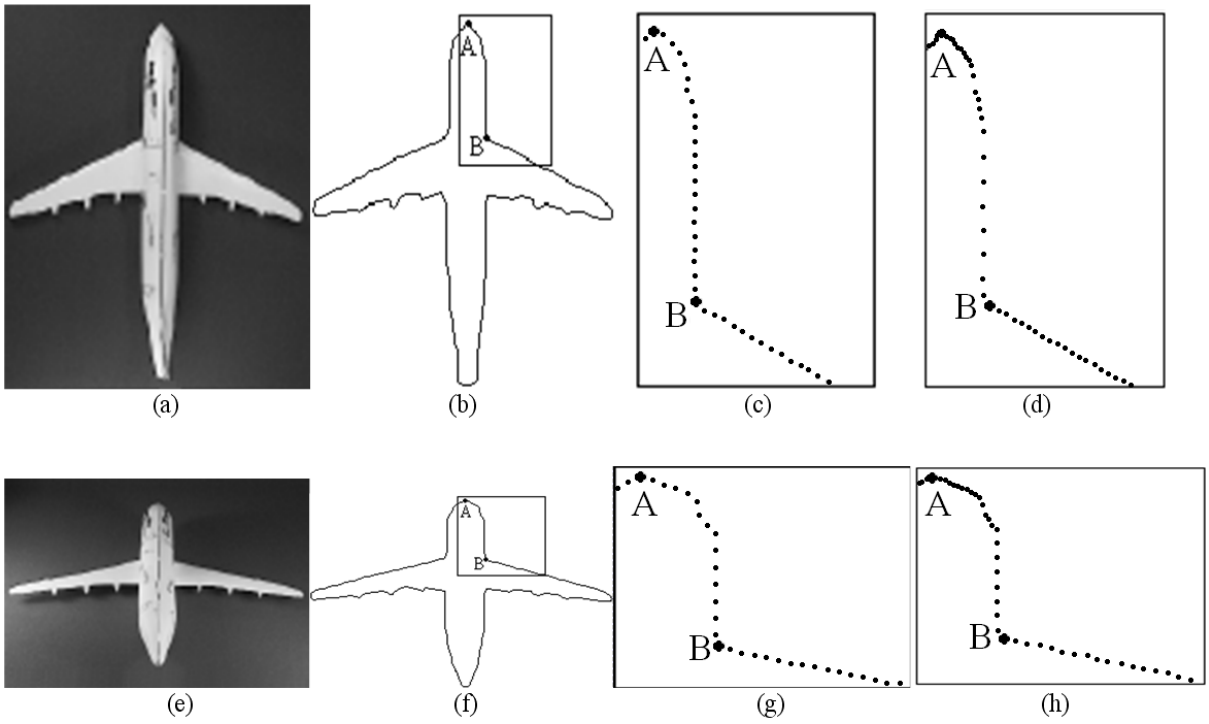
where $x(s)$ and $y(s)$ are the coordinates of points on the contour with the origin of the system located at the centroid of the contour and α and β the curvilinear abscissa of two points on it.

The parameter ψ is essentially the cumulative sum of triangular areas produced from connecting the centroid to pairs of successive vertices on the contour.

The two parameters, τ and ψ, are linear under a general affine transformation, since they are scaled with respect to their images under an affine transform. They can be made completely invariant by simply normalizing them with respect to either the total affine arc length or the enclosed area of the contour. So, both the affine invariant parameters can be used to parametrize object contours. However, they have different properties and can be chosen based on the application.

The first and second derivatives are required in the computation of the affine arc length τ. It is well known that the derivatives are very sensitive to noise. Therefore, if the contours are heavily corrupted by noise, a smoothing process must be performed before parametrizing the contours, using the affine arc length as a parameter.

In the second parameter, the centroid of the contour must be used as the origin of the system in order to make the parametrization translation invariant. As a result, this parameter can be used only if the contours are closed. It cannot be used in the case of occulted objects.



(a) The image of the top view of a plane; (b) the contour of image (a); (c) a part of contour (b) normalized by equidistant vertices; (d) a part of contour (b) normalized by iso-area; (e) the image of rear top view of the plane; (f) the contour of the image (e); (g) a part of contour (f) normalized by equidistant vertices; (h) a part of contour (f) normalized by iso-area.

Figure 3.1: The comparison of equidistant vertices normalization and iso-area normalization

3.3 Iso-area normalization

All points on a contour could be expressed in terms of the parameter of index points along the contour curve starting from a specified point. With affine transforms, the position of each point

changes and it is possible that the number of points between two specified points changes too. So if we parametrize the contour using the equidistant vertices, the index of point along the contour curve will change under affine transforms. For example, Figure 3.1(a) is the top view of a plane, and (e) is its rear top view, so (e) can be considered as one of possible affine transforms of image (a). Via region segmentation or edge following, we obtain the contours of the two plane views (b) and (f). (c) and (g) are parts of the contours (b) and (f) normalized by equidistant vertices respectively. In Figure 3.1(c), the number of points on the segment between the points A and B is 21; however, the number is 14 in the same segment in Figure 3.1(g). So the contour normalized by equidistant vertices is variant under possible affine transforms.

In order to make it invariant under affine transforms, a novel curve normalization approach is proposed. This provides an affine invariant description of object curves at low computational cost, while at the same time preserving all information on curve shapes. We call this approach as iso-area normalization (IAN).

All points on a shape contour could be expressed in terms of the function , $\hat{\Gamma}(m) = (\hat{x}(m), \hat{y}(m))$, $m \in [0,\ M-1]$ where variable m is measured along the contour curve from a specified starting point. M is the total number of points on the contour. The steps of IAN are presented as follows:

1. Normalize to N points with equidistant vertices. The new function is $\bar{\Gamma}(\mu) = (\bar{x}(\mu), \bar{y}(\mu))$ and all the points on the contour are $\bar{P}_\mu$, where $\mu \in [0,\ N-1]$. Point $\bar{P}_N(\bar{x}(N), \bar{y}(N))$ is assumed to be the same as the first point $\bar{P}_0(\bar{x}(0), \bar{y}(0))$.

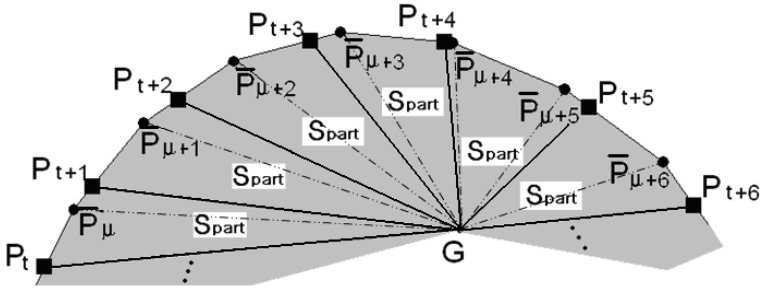
2. Calculate its centroid G using Eq. 1.9.

3. Transfer the contour to make its centroid G the origin of the coordinates system.

4. Compute the area of the contour using the formula:

$$S = \frac{1}{2} \sum_{\mu=0}^{N-1} |\bar{x}(\mu)\bar{y}(\mu+1) - \bar{x}(\mu+1)\bar{y}(\mu)| \tag{3.5}$$

   where $\frac{1}{2}|\bar{x}(\mu)\bar{y}(\mu+1) - \bar{x}(\mu+1)\bar{y}(\mu)|$ is the area of the triangle whose vertices are centroid G, $\bar{P}_\mu(\bar{x}(\mu),\ \bar{y}(\mu))$ and $\bar{P}_{\mu+1}(\bar{x}(\mu+1),\ \bar{y}(\mu+1))$ (cf. Figure 3.2).



Dots (•) are the vertex of equidistant vertices normalization, and squares (■) are the point P of iso-area normalization. G is the centroid of the contour.

Figure 3.2: The method of iso-area normalization

5. Let the number of points on the contour after IAN also be N, of course, any other number of points could be chosen. Therefore, after IAN, each enclosed area S_{part} defined by any two successive points on the contour and the centroid G is equal to $S_{part} = S/N$.

6. Suppose all the points on the contour after IAN are P_t. Let $\Gamma(t) = (x(t), y(t))$ represent the corresponding contour, where $t \in [0,\ N-1]$. Select point $\bar{P}_0(\bar{x}(0),\ \bar{y}(0))$ on the equidistant vertices normalization as the starting point $P_0(x(0),\ y(0))$ of the IAN. On segment $P_0\bar{P}_1$, we seek a point $P_1(x(1), y(1))$, so that the area $s(0)$ of the triangle whose vertices are $P_0(x(0), y(0))$, $P_1(x(1), y(1))$ and $G(0, 0)$ is equal to S_{part}. If there is no point to satisfy this condition, we seek the point P_1 on the segment $\bar{P}_1\bar{P}_2$. So area $s(0)$, which is the sum of the areas of triangle $\bar{P}_0\bar{P}_1G$ and triangle $\bar{P}_1P_1G$, is equal to S_{part}. If again there is yet no point to satisfy the condition, we continue to seek for the point in the next segment until the condition is satisfied. This point P_1 is the second point on the iso-area normalized contour.

7. From point $P_1(x(1), y(1))$, we use the same method to calculate all the other points $P_t(x(t), y(t))$, $t \in [2,\ N-1]$ along the contour. Because the area of each enclosed zone, for example, the polygon $P_t[\bar{P}_\mu\bar{P}_{\mu+1}\cdots]P_{t+1}G$, $t \in [0,\ N-2]$ is equal to S_{part}, the total area of $N-1$ polygon is $(N-1)\cdot S_{part}$. According to the step 5, the area $s(N-1)$ of the last zone $P_{N-1}[\bar{P}_\mu\bar{P}_{\mu+1}\cdots\bar{P}_{N-1}]P_0G$ is exactly equal to:

$$s(N-1) = S - (N-1)\cdot S_{part} = N\cdot S_{part} - (N-1)\cdot S_{part} = S_{part}$$

From Figure 3.2 we know that the area of triangle $P_tP_{t+1}G$ is approximately equal to the area S_{part} of polygon $P_t[\bar{P}_\mu\bar{P}_{\mu+1}\cdots]P_{t+1}G$ if the two points $\bar{P}_\mu$ and $\bar{P}_{\mu+1}$ are close enough or the number N of the points on the contour is large enough. Therefore, we can use the points P_t, $t \in [0,\ N-1]$ to replace the points $\bar{P}_\mu$, $\mu \in [0,\ N-1]$. The contour with the vertices P_t is the contour normalized by IAN.

According to subsection *3.2*, after this normalization, the number of vertices on the segment between two appointed points is invariant under affine transforms. Figure 3.1(d) and (h) are the same parts of Figure 3.1(c) and (g), respectively. We notice that the distance between the consecutive points is not uniform. In Figure 3.1(d), the number of points between points A and B is 23, this number is also 23 in Figure 3.1(g). Therefore, after applying IAN, the index of the points on a contour can remain stable with their positions under affine transforms. This property will be very advantageous when extracting the robust attributes of a contour and decreasing complexity in the measurement of similarity.

We can also use IAN with other algorithms, to improve their robustness against affine transforms. For example, before applying the curvature scale space (CSS) algorithm [45], the contour can be normalized by IAN: Figure 3.3 shows an example. (a) and (d) are visualized original contours normalized by equidistant vertices and by IAN; (b) and (e) are affine transformed contours of (a) and (d) normalized by equidistant vertices and by IAN; (c) are CSS images of (a) and (b); (f) are CSS images of (d) and (e). In these CSS images the light lines are the CSS images of original contours; the bold lines are the CSS images of affine transformed contours. σ is the standard deviation of Gaussian filter and μ is the index of the points on the shapes. We notice that, under affine transform, the CSS image extracted from the contour normalized by IAN is more stable than that coming from the contour normalized by equidistant vertices. Clearly, this is beneficial when calculating the similarity between two CSS attributes.

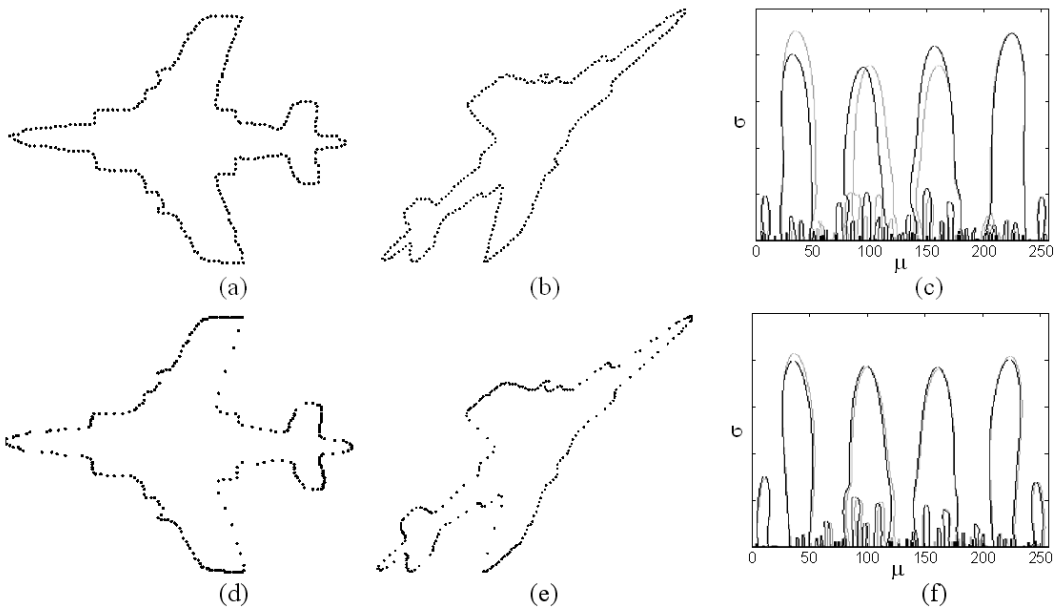


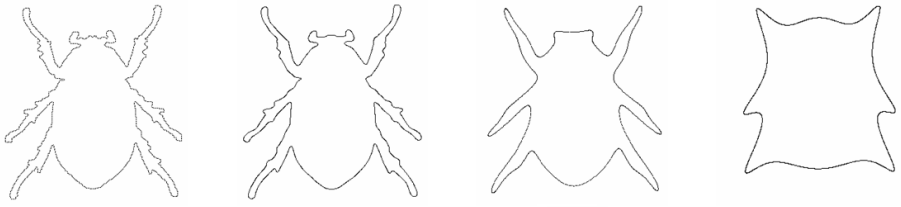
Figure 3.3: The comparison of CSS images coming from the contours normalized by equidistant vertices and by IAN

3.4 Scale-controlled area-difference shape descriptor (SCAD)

In this section, we propose an IAN-based shape representation and description well adapted to pattern recognition, particularly in the context of affine shape transformations. The proposed approach operates from a single closed contour. The comparison result of two scale-space contours forms a shape descriptor. For more detail information, please refer to our paper [77].

3.4.1 Inspiration of psychophysics

A multiscale shape representation of an object where the shape variations are expressed with respect to a given scale, provides different information about the object. We take Figure 3.4 as an example.



(a) Beetle original contour; (b) (c) and (d) are higher scale representations of the beetle.

Figure 3.4: Multiscale shape representation

When we observe the Figure 3.4(b) or (c), without doubt, it is also a beetle. It shows that although (a) and (b), or (a) and (c) are not exactly the same, (b) and (c) contain the main information that is needed to distinguish the beetle. This so-called main information

is the essential segment information. The human vision distinguishes the object only from characteristics of main segments [97]. These segment characteristics include the segment length, the segment curving degree and segment curving direction, etc.

Therefore, extracting correctly the information of the original shape as essential segment, expressing it with the reasonable mathematical format and enabling different shapes to be compared quantitatively, are the key objects of this algorithm.

As introduced in Section 2.6, in the curvature scale space (CSS) approach, the Gaussian filtering and the computation of curvature zero-crossing points are the basic algorithms. The CSS image consists of several arches, each related to a concavity or a convexity of the curve. In many applications, the maxima of these arches are used to represent the contour of an object. To construct the CSS image of a digital curve, its curvature zero-crossing points should be determined at different scales of smoothing. In Gaussian smoothing, there will be two curvature zero-crossing points on every concave or convex part of the shape and as the curve becomes smoother these points approach each other. The locations of each zero-crossing at different levels of scale create an arch in the CSS image. At a certain level of smoothing when the segment is filled, the two points join and represent the maximum of the relevant arch. The height of this arch then reflects the depth and size of the concavity or convexity. The deeper and larger a segment is, the higher is the maximum. The resulting shape descriptor is very compact and extensive tests [98] reveal that the method is quite robust with respect to noise, scale and orientation. Because of these advantages, the CSS shape descriptor has been included as one of the two shape descriptors in the ISO/IEC MPEG-7 standard [8].

As shown in [99], however, it does not always give results in accordance with human vision system. The main drawbacks of this description are due to the problem of shallow concavities/convexities on the shape. It can be shown that the shallow and deep concavities/convexities may create the same large arches on the CSS image. Therefore, a shallow concavity/convexity may be matched with a deep one during the CSS matching [76].

The analysis of the ambiguities of CSS clearly shows that the algorithm only focuses to the position of the maxima of the inflexion points, but it neglects curvature strength, which is the essential characteristic segment between both inflexion points. Therefore, it cannot distinguish between a shallow concavity with a deep one.

We know that the curved segments and the straight lines are the basic essential segments of a shape [5, 100]. If we can divide a shape reasonably into several segments and extract these basic essential segments as the attributes from each one, using these attributes as shape descriptors will be reasonable. This consideration is on a shape characterization point of view. Based on the multiscale analysis, dividing the essential segments and extracting the description of attributes essential segments on the shape will be expatiated on the following section.

The so-called essential segment is situated between two key points in the shape contour. Obviously, in order to represent the entire shape with the essential segments, the selection of key points is very important. In other words, the selection of key points is not the final goal; its goal is to extract the attribute of essential segments that are located between each pair of key points on the contour. Each contour consists of several essential segments. These essential segments should have the following properties:

- Property 1: each essential segment should denote the macroscopic attribute of a part of the shape, but not the fine details.

- Property 2: the sign of curvatures of most points on an essential segment is consistent.

- Property 3: the order of each essential segment on a shape contour plays a vital role for describing the whole shape.

From an analysis based on above properties, we use higher scale to simplify the shape and to retain the macroscopic part of original shape, thus we obtain Property 1. In order to obtain the curving direction of each segment, we may compute the curvature of each segment, thus obtaining Property 2. In the proposed descriptor, the order of succession of each essential segment on a shape contour is one of the retrieving attributes of the shape, namely the Property 3.

We assume that shapes have already been extracted from images in the form of closed sequences of points. The proposed approach consists in 5 steps: 1) shape IAN normalization, 2) scale-controlled filtering with the threshold of segment-length, 3) short segments merging or removing, 4) scale-controlled shapes registration, 5) extraction of essential segment attributes.

IAN normalization as a general method have been explained in Section 3.3. We will introduce the last 4 steps in details as the followings.

3.4.2 Scale-controlled filtering

In order to remove the fine-drawn details in the shape, and only retain the main body of shape that takes part in the main function for shape retrieval, we use a filter to simplify the shape. Both Gaussian filtering and Fourier transform may achieve this goal [101]. Here, a delicate task is selecting the right threshold for stopping filtering and thus defining the level of shape simplification. If high frequency components are filtered slightly, see Figure 3.4(b), then the change of details in the shape may affect, in some particular context, the result of the machine recognition. Otherwise, if the high frequency components are filtered highly, see Figure 3.4(d), then the shape distorts very much, so that it becomes hard to recognize the original shape. The experimental studies indicate that choosing an appropriate threshold, based on observations of a minimal segment-length between two curvature zero-crossing points, is a good basis for defining the strength of filtering, and allows extraction of an appropriate shape.

As introduced in CSS, with σ decreasing, the number of curvature zero-crossing points increases: the segment-length between the curvature zero-crossing points goes shorter and the shape distorts more slightly and vice versa. In other words, the threshold of segment-length fixes the distortion degree of the original shape.

Regarding a shape S, the stopping condition for filtering is that there are not two consecutive segments that are shorter than a threshold L. If this condition cannot be reached, then increase σ once more and filter again until the condition is matched. In order to reduce the computational complexity, we use a dichotomic approach. The flow chart of the algorithm is shown on Figure 3.5. It also includes a limit threshold on weak variance of σ.

To compare the computational complexity of SCAD with CSS in multiscale filtering, suppose the precision of filter scale is $\Delta\sigma$, use the dichotomic approach, when the condition of segment-length threshold is reached, σ is between minimum σ_B and maximum σ_G, then the number of filterings and the curvature computing is

$$T_s = 1 + \log_2(\sigma_G/\Delta\sigma)$$

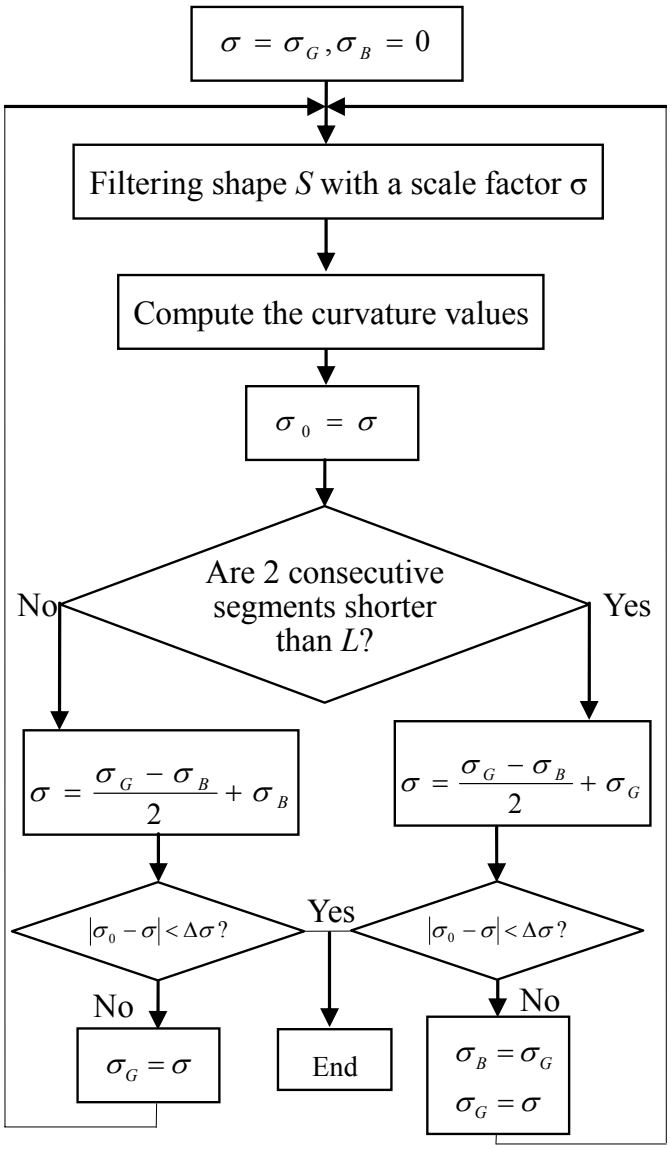


Figure 3.5: The flowchart of dichotomic approach

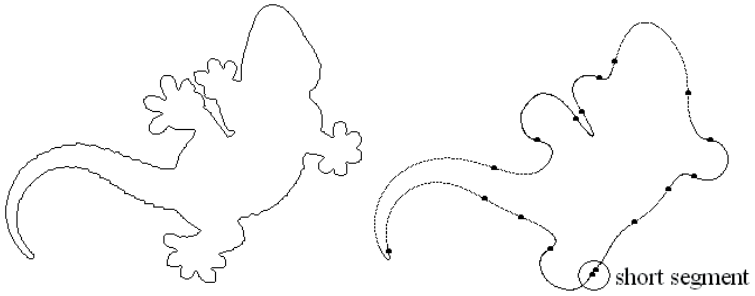
In the traditional CSS method, in order to obtain the CSS picture, the filter scale needs to increase gradually with the precision $\Delta\sigma$, then the maximum number of filtering and the curvature computing is:

$$T_{Cmax} = \sigma_G/\Delta\sigma$$

For example, if $\Delta\sigma = 0.125$, $\sigma_G = 64$, $\sigma_B = 0$, then $T_s = 10$, $T_{Cmax} = 512$.

Therefore, dichotomic approach may increase greatly the computation efficiency.

After scale-controlled filtering with the threshold of segment-length and curvature computing, we obtain a simplified shape composed by the basic essential segments, see Figure 3.6.



(a) The original contour; (b) the scale-controlled filtered contour. The dots (•) on the contour are the curvature zero-crossing points.

Figure 3.6: The examples of result of scale-controlled filtering

3.4.3 Short segments merging and removing

In Figure 3.6, we take notice of the short segments on the contour. Clearly, the short segments cannot represent the main characteristic of a shape. In Subsection 3.4.2, in order to maintain the main characteristics of a shape, we use the threshold condition that two consecutive segments are shorter than a threshold. Therefore, it is possible that some independent short segments remain on the shape. These short segments may affect the result of the machine recognition. According to the states of essential segment Property 1, the short segments must be removed or merged.

The removing and merging algorithms are as follows:

Chord AD connects two key points A, D, which are the neighborhood points of short segment BC. If all the points on the segment AD are on the same side of the chord AD, as shown on Figure 3.7(a), the key points B and C are deleted. Thus, the three segments a, b and c will merge into the long segment m, see Figure 3.7(b). If the points on the segment AD are located on the two sides of chord AD, Figure 3.7(c), take the middle point E between the points B and C to replace them. Thus, the three segments a, b and c merge into two longer segments m and n, Figure 3.7(d).

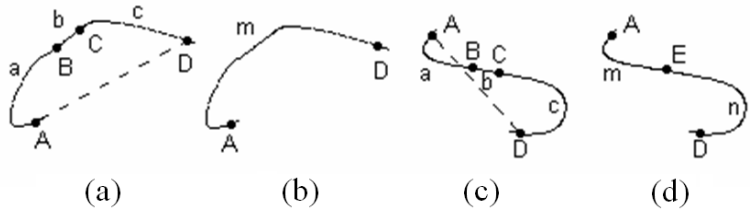


Figure 3.7: Example of short segments merging and removing

After this step, Figure 3.6(b) turns into Figure 3.8. Notice that there are not very short segments on the contour.

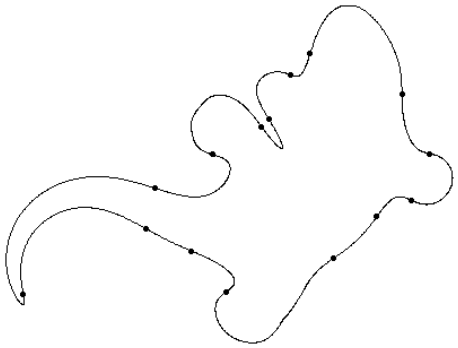


Figure 3.8: The filtered contour Figure 3.6(b) after merging or removing short segment

3.4.4 Essential segments registration

After the previous steps, the shape is transformed from the original shape that had rich details to the main outline only. We superpose Figure 3.6(a) and Figure 3.8 in the same picture and mark the key points on the original and filtered shape, as shown on Figure 3.9.

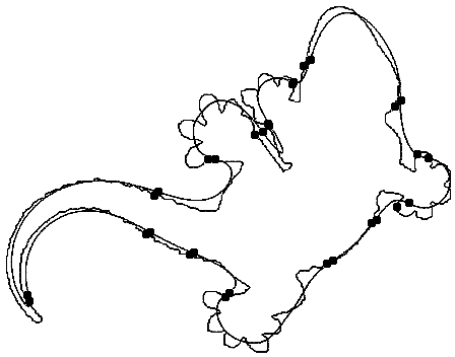


Figure 3.9: The contours of Figure 3.6(a) and Figure 3.8 drawn together

After Gaussian filtering, the coordinate of contour points may be displaced. In order to compare the changes between the two contours, at first, we carry on essential segments registration. Because the change degree of each point after filtering is not the same, therefore the process of registration is realized segment by segment.

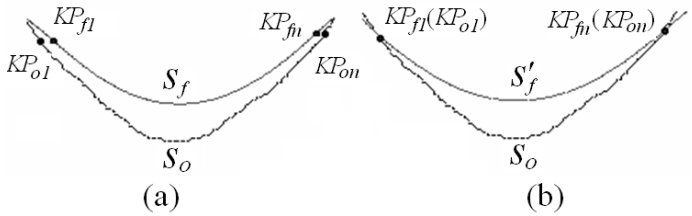


Figure 3.10: Scale-controlled essential segments registration

On Figure 3.10(a), s_o is a segment on an original contour, s_f is the corresponding segment on the filtered contour. KP_{o1}, KP_{on} are the key points on the original contour, and their respective

coordinates are $(x_{o1},\ y_{o1})$, $(x_{on},\ y_{on})$; $KP_{f1},\ KP_{fn}$ are the corresponding key points on the filtered contour, their respective coordinates are (x_{f1}, y_{f1}), $(x_{fn},\ y_{fn})$; there are n points on both segment s_o and s_f. On the filtered contour, the points set between KP_{f1} and KP_{fn} is

$$M = \{P_i(x_i, y_i),\ i = 1, 2, \cdots, n\}$$

The registration algorithm from segment s_f to segment s_o is as follows:

$$\begin{cases} \Delta x_1 = x_{f1} - x_{o1} \\ \Delta y_1 = y_{f1} - y_{o1} \end{cases} \quad and \quad \begin{cases} \Delta x_n = x_{fn} - x_{on} \\ \Delta y_n = y_{fn} - y_{on} \end{cases}$$

Let $M' = \{P'_i(x'_i, y'_i),\ i = 1, 2, \cdots, n\}$ be the set of registered points, so

$$\begin{cases} x'_i = x_i - (\Delta x_1 \frac{n-i}{n} + \Delta x_n \frac{i}{n}) \\ y'_i = y_i - (\Delta y_1 \frac{n-i}{n} + \Delta y_n \frac{i}{n}) \end{cases}$$

After registration, the segment s_f in Figure 3.10(a) turns to the segment s'_f in (b). The key points KP_{o1} and KP_{f1} superpose, KP_{on} and KP_{fn} are same.

Figure 3.11 shows the contours in Figure 3.9 of original and filtered registering. We denote each essential segment on the contour, namely $s_1, s_2, \cdots, s_{15}$ respectively.

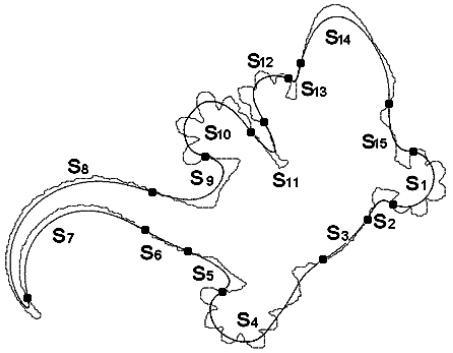


Figure 3.11: The original and filtered contours after registering

3.4.5 Extraction of essential segment attributes

According to different purposes, we can extract different attributes from the essential segments. These attributes include the segment-length, the rotation angle of the segment [100], the direction change between the neighboring essential segments, the average (or the highest) curvature of a segment, the segment deepness and so on. We will use the multiscale characteristic and propose the method for extracting the attributes that are robust against affine transformation.

As Figure 3.11 shows, the effect of filter is different on different essential segments. The segments with details, for example, s_1, s_4 and s_{10}, change more than the smooth segments after filtering. While the smoothest segments on the original shape, for example, s_3 and s_6 change slightly after the filter. The cause is that the low-pass filter affects high frequency components more than low frequency components. In addition, the geometric heat equation, which is a kind of curvature deformation, clearly shows that the intensity of movement of a point on a segment is proportional to the value of curvature at that point [102]. Measuring and expressing the change degree is the content of scale-controlled area-difference shape descriptor (SCAD). To quantify

the change degree, we detail the algorithm as follows: in Figure 3.12, the two segments C_1 and C_2 come from the original and filtered contours respectively. Because they are between pairs of corresponding zero-crossing points, the number of points on both segments is always the same, so we can calculate the area between both segments point by point.

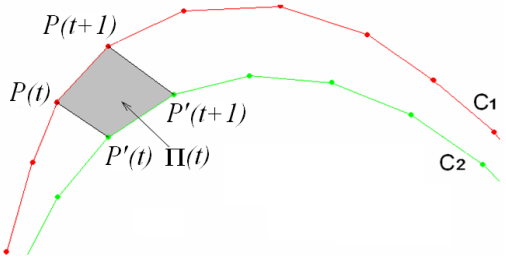


Figure 3.12: The area between original and filtered segments

Let P be a point on the original contour and P' the corresponding point on the filtered contour. If the coordinates of the points $P(t)$, $P(t+1)$, $P'(t)$ and $P'(t+1)$ are $(x(t), y(t))$, $(x(t+1), y(t+1))$, $(x'(t), y'(t))$ and $(x'(t+1), y'(t+1))$ respectively, therefore, the area $\Pi(t)$ defined by the 4 points is:

$$\begin{aligned}\Pi(t) &= \frac{1}{2} \mid [x(t)y(t+1) - x(t+1)y(t)] + \left[x(t+1)y'(t+1) - x'(t+1)y(t+1)\right] \\ &\quad + \left[x'(t+1)y'(t) - x'(t)y'(t+1)\right] + \left[x'(t)y(t) - x(t)y'(t)\right] \mid\end{aligned}$$

The SCAD value of the j^{th} segment can now be defined as:

$$v_j = \frac{1}{L_j} \sum_{t=t_j}^{t_{j+1}-1} \Pi(t)$$

where t_j is the starting point on the j^{th} segment and t_{j+1} is the end point on it; L_j is the arc length of the j^{th} filtered segment.

Then we define the SCAD-based descriptor as:

$$T_{SCAD} = \{(n_j, v_j)\,,\ j = 0, 1, 2, \cdots, m-1\}$$

where m is the number of essential segments on the shape and n_j is the curvilinear abscissa of the j^{th} essential segment.

Figure 3.13 represents the SCAD-based descriptor of the shape Figure 3.6(a), where $s_1, s_2, \cdots, s_{15}$ are the corresponding essential segments. (n_1, v_1) and (n_{10}, v_{10}), for example, are the descriptor of segments s_1 and s_{10} respectively.

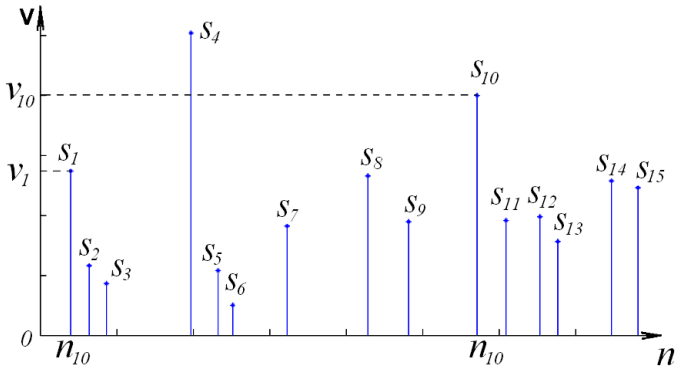


Figure 3.13: The SCAD-based descriptor of Figure 3.6(a)

3.4.6 Properties of SCAD

Each key contour points defined by the SCAD algorithm, is obtained by the Gaussian filtering and computation of curvature zero-crossing points. It is quite the same as the CSS algorithm. The SCAD algorithm yields the same important properties that the CSS algorithm has [76]:

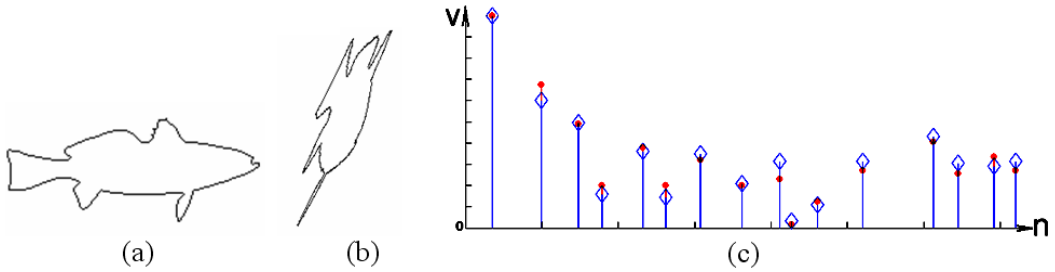
1. it reflects properties of the human visual system perception, thus offering good generalization.

2. it is robust to noise and changes in scale and orientation.

3. it is compact, reliable and fast.

To create the CSS image, the scale σ increases constantly, until all the curvature zero-crossing points on the shape vanish. This increases the computation time. For SCAD, as the filter scale is decided by the segment-length between the curvature zero-crossing points, we can control the contour to retain appropriate shape and reduce the computational complexity.

Furthermore, the SCAD is robust with affine transforms. Figure 3.14 shows an example that the SCAD representations have a great correlation between the original and the affine transformed one.



(a) Original contour; (b) one of its affine transformed contour; (c) their SCAD-based descriptors. Diamond (◇) represents the SCAD points of original contour and dot (•) represents the SCAD points of affine transformed contour.

Figure 3.14: SCAD-based descriptor of the original contour and its affine transform

3.4.7 Conclusion about SCAD

The proposed technique, SCAD, makes use of the essential segments of contour coordinates at appropriate scale levels. The scale level is decided according to the shortest arc-length on a contour. The result of comparing every original segment with the corresponding filtered segment defines the SCAD descriptor. This new approach can be used to retrieve objects undergoing an affine transformation. Moreover the proposed method has the supplementary advantage of a low computational complexity.

3.5 Enclosed area affine invariant function descriptor

In this section, for the two-dimensional object silhouettes, we present a one-dimensional descriptor which in theory remains absolutely invariant under affine transforms. The proposed descriptor operates on iso-area normalization. We prove that for any linearly filtered contour, the area of a triangle, whose vertices respective are the centroid of the contour and a pair of successive points on the normalized contour, remains linear under affine transforms. Experimental results indicate that the proposed method is unaffected by boundary starting point variations and affine transforms even in the case of high deformations and serious noise contamination on the shapes. For more detail information, please refer to our paper [103].

3.5.1 Normalized partial area vector (NPAV)

To extract an affine invariant attribute, we begin with an iso-area normalized contour (Section 3.3) and look for the existing relations between the partial area S_{part}, affine transforms and low-pass filtering.

- THEOREM1:

If $\Gamma_a(\mu) = (x_a(\mu), y_a(\mu))$ is the transformed version of a curve $\Gamma(\mu) = (x(\mu), y(\mu))$ under an affine transform A, where μ is an arbitrary parameter, $\Gamma_{af}(\mu) = (x_{af}(\mu), y_{af}(\mu))$ notes that $\Gamma_a(\mu)$ is filtered by a linear low-pass filter F. If $\Gamma_f(\mu) = (x_f(\mu), y_f(\mu))$ notes that $\Gamma(\mu)$ is filtered by the same low-pass filter F, $\Gamma_{fa}(\mu) = (x_{fa}(\mu), y_{fa}(\mu))$ refers to the transformed version of $\Gamma_f(\mu)$ under the same affine transform A. The curve $\Gamma_{af}(\mu)$ is then the same as curve $\Gamma_{fa}(\mu)$. In other words: $F(A(\Gamma(\mu))) = A(F(\Gamma(\mu)))$ (cf. Figure 3.15).

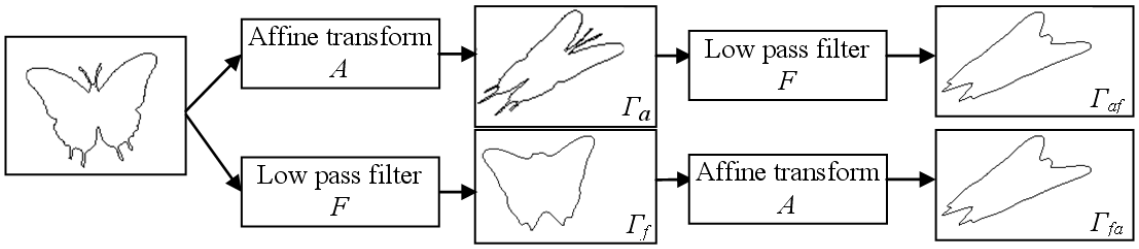


Figure 3.15: Illustration of theorem1

PROOF:

For the entire contour, we transfer its center of gravity to the origin of the system. So in affine transform Eq. 3.1, the translation e and f can be removed. Therefore, the affine transform can be represented by two simple formulae:

$$\begin{cases} x_a(\mu) = ax(\mu) + by(\mu) \\ y_a(\mu) = cx(\mu) + dy(\mu) \end{cases} \tag{3.6}$$

The computation starts by convolving each coordinate of the curve $\Gamma_a(\mu)$ with a linear low-pass filter F whose impulse response is $g(\mu)$. In the continuous form this leads to:

$$\begin{aligned} x_{af}(\mu) &= x_a(\mu) * g(\mu) = [ax(\mu) + by(\mu)] * g(\mu) \\ &= ax(\mu) * g(\mu) + by(\mu) * g(\mu) = ax_f(\mu) + by_f(\mu) \end{aligned} \tag{3.7}$$

where $*$ denotes the convolution. Likewise

$$y_{af}(\mu) = cx_f(\mu) + dy_f(\mu) \tag{3.8}$$

From a comparison of Eq. 3.7, 3.8 and Eq. 3.6, it is clear that point $(x_{af}(\mu), y_{af}(\mu))$ is the same as point $(x_f(\mu), y_f(\mu))$ transformed by the affine transform A. So curve $\Gamma_{af}(\mu)$ is the same as curve $\Gamma_{fa}(\mu)$. Theorem1 indicates that exchanging the computation order between affine transform and filtering does not change the result.

- THEOREM2:

For any affine transform of a closed contour, using IAN sets parameter t to produce the curve $\Gamma_a(t) = (x_a(t), y_a(t))$. If area $s_p(t)$ is the area of an enclosed sector whose vertices are a pair of successive points and the centroid of the contour and if $\Gamma_{af}(t) = (x_{af}(t), y_{af}(t))$ indicates that $\Gamma_a(t)$ is filtered by a low-pass filter F, then the changes in enclosed areas $s_p(t)$ on the $\Gamma_{af}(t)$ are linear with affine mapping. See Figure 3.16.

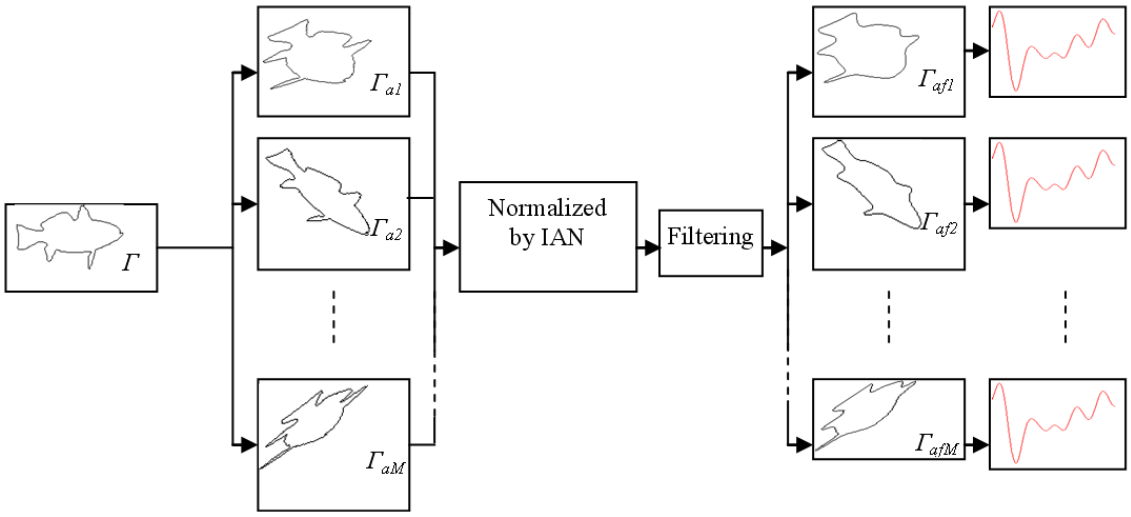


Figure 3.16: Illustration of theorem2

PROOF:

From section 3.3, we know the enclosed area $s_p(t)$ of the triangle on the filtered affine contour whose vertices are $(x_{af}(t), y_{af}(t))$, $(x_{af}(t+1), y_{af}(t+1))$ and the centroid G is

$$s_p(t) = \frac{1}{2} \left| x_{af}(t) y_{af}(t+1) - x_{af}(t+1) y_{af}(t) \right| \tag{3.9}$$

Due to THEOREM1,

$$\begin{cases} x_{af}(t) = x_{fa}(t) = ax_f(t) + by_f(t) \\ y_{af}(t) = y_{fa}(t) = cx_f(t) + dy_f(t) \end{cases}$$

and

$$\begin{cases} x_{af}(t+1) = x_{fa}(t+1) = ax_f(t+1) + by_f(t+1) \\ y_{af}(t+1) = y_{fa}(t+1) = cx_f(t+1) + dy_f(t+1) \end{cases}$$

Therefore from Eq. 3.9,

$$\begin{aligned} s_p(t) &= \tfrac{1}{2} \mid [ax_f(t) + by_f(t)]\,[cx_f(t+1) + dy_f(t+1)] \\ &\quad - [ax_f(t+1) + by_f(t+1)]\,[cx_f(t) + dy_f(t)] \mid \\ &= \tfrac{1}{2} \mid adx_f(t)y_f(t+1) + bcx_f(t+1)y_f(t) \\ &\quad -adx_f(t+1)y_f(t) - bcy_f(t+1)x_f(t) \mid \\ &= \tfrac{1}{2}\,|ad - bc| \cdot |x_f(t)y_f(t+1) - x_f(t+1)y_f(t)| \end{aligned} \tag{3.10}$$

Observing Eq. 3.10, $s_p(t)$ is just linearly proportional by a scale factor $|ad - bc|$. Accordingly, we have proved that enclosed areas $s_p(t)$ are linear with affine mapping.

- DEDUCTION:

The proportion $v'(t)$ of enclosed areas $s_p(t)$ with the total area S of the filtered contour is preserved under general affine transforms.

PROOF: According to Eq. 3.10, the total area S of the filtered contour is:

$$S = \frac{1}{2}\,|ad - bc| \cdot \sum_{t=0}^{N-1} |x_f(t)y_f(t+1) - x_f(t+1)y_f(t)|$$

So

$$\begin{aligned} v'(t) &= s_p(t)/S \\ &= |x_f(t)y_f(t+1) - x_f(t+1)y_f(t)| \,/\, \textstyle\sum_{t=0}^{N-1} |x_f(t)y_f(t+1) - x_f(t+1)y_f(t)| \end{aligned} \tag{3.11}$$

Eq. 3.11 indicates that $v'(t)$ is not related to the affine parameters *a, b, c* and *d*. Therefore $v'(t)$ is preserved under general affine transforms.

We refer to vector

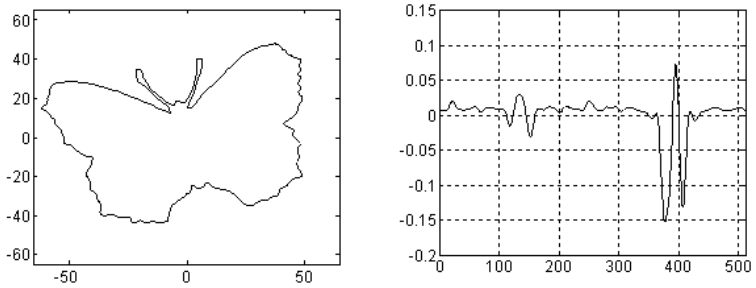
$$v(t) = \left[v'(t) - 1/N\right] \tag{3.12}$$

as the normalized partial area vector (NPAV). We can deduce a major property of $v(t)$: the integration of $v(t)$ is equal to zero.

PROOF:

$$\sum_{t=0}^{N-1} v(t) = \sum_{t=0}^{N-1} \left[v'(t) - \frac{1}{N}\right] = \sum_{t=0}^{N-1} \frac{s_p(t)}{S} - \sum_{t=0}^{N-1} \frac{1}{N} = \frac{S}{S} - \frac{N}{N} = 0$$

Figure 3.17 is an example of NPAV. The contour is normalized to 512 points by IAN.



(a) The contour of a butterfly; (b) the NPAV of the contour (a).

Figure 3.17: An example of NPAV

As theorem2 and its deduction show that, in all cases, even those with severe affine transformation, the function $s_p(t)$ is also preserved. Only the amplitude changes under general affine transforms; the NPAV $v(t)$ has an affine-invariant feature. In the following section, we will present the results of our experiments which evaluate the property of the proposed algorithm.

3.5.2 Experimental results of NPAV

In this section, we will evaluate the behavior of NPAV $v(t)$ in relation to affine transforms, IAN and noise by presenting various experimental results. We consider the results of our experiments on the MPEG-7 CE-shape-1 database which consists of 1400 shapes semantically classified into 70 classes [104] and the database of 20 planes that is presented in [16].

To measure the similarity between two NPAVs, we use circular correlation function. The circular correlation function is defined by:

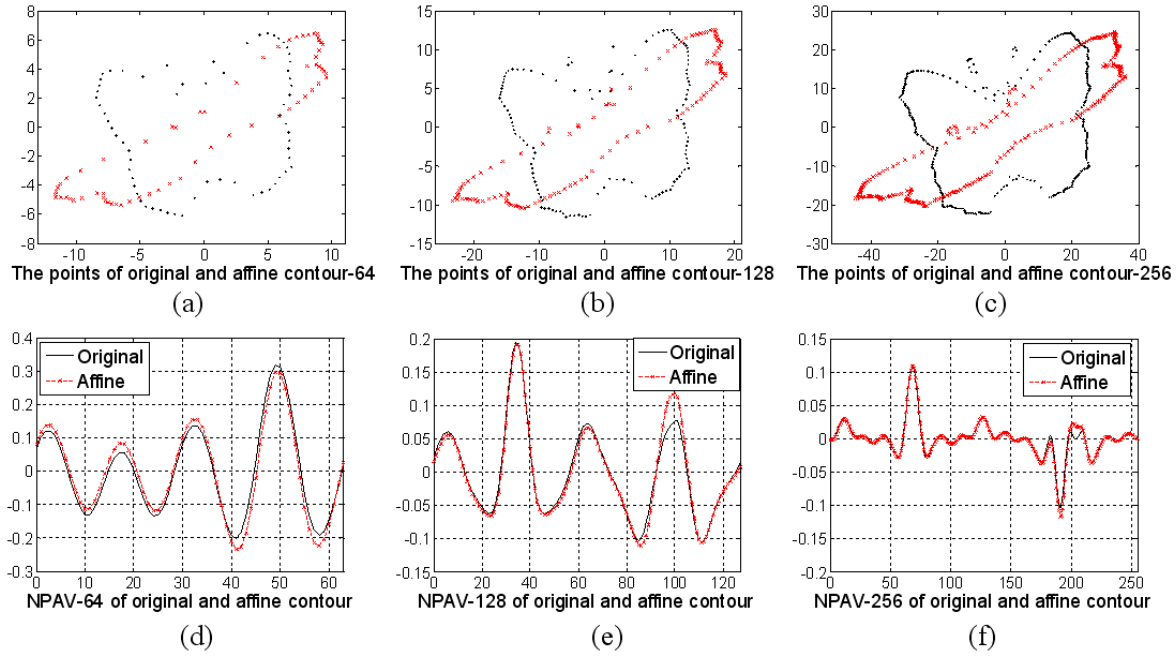
$$R(v_1(t), v_2(t)) = \max_{n=0}^{N-1} \left[\frac{\sum_{t=0}^{N-1} v_1(t) v_2(s)}{\|v_1\| \cdot \|v_2\|} \right], \quad s = mod(t+n, N) \tag{3.13}$$

where $\|\bullet\|$ is the norm; $v_\alpha(t), \alpha = 1, 2$, are two NPAVs of contours as defined by Eq. 3.12.

3.5.2.1 NPAV and number of points normalized by IAN

We first test the effect of different number of points normalized by IAN to NPAV on MPEG-7 CE-Shape-1 database. The number of points on the contour is respectively normalized to 64, 128 and 256. In this experiment, we use Gaussian low-pass filter with standard deviation $\sigma = 10$. As we know, for a same low-pass filter, the shorter a signal is, the lower the relative band width is, i.e., the more high-frequencies are filtered.

Figure 3.18 shows the original and the affine contours of the pattern in Figure 3.17(a) with various numbers of points and their NPAVs. We notice that for the contour with lesser points, its NPAV function is smoother, and vice versa.



(a)-(c) are the original and affine contours normalized to 64, 128 and 256 respectively; (d)-(f) are the NPAVs of the two contours in (a)-(c) respectively.

Figure 3.18: Illustration of the robustness of NPAV under the contour normalized to different number of points

Although the NPAVs of the contour with different numbers of points are very different from each other, under the same number of points normalized by IAN the NPAV of the affine contour and that of original contour are almost identical. So under affine transform, NPAV remains invariant with different number of normalized points on a contour.

Furthermore, we calculate the statistic results. For each shape in MPEG-7 CE-shape-1 database, the average correlations between the NPAV of the original shape and that of its affine transform and this under the number of points 64, 128 and 256 are presented in Table 3.1.

The number of points	Average correlation coefficient
64	0.985
128	0.992
256	0.993

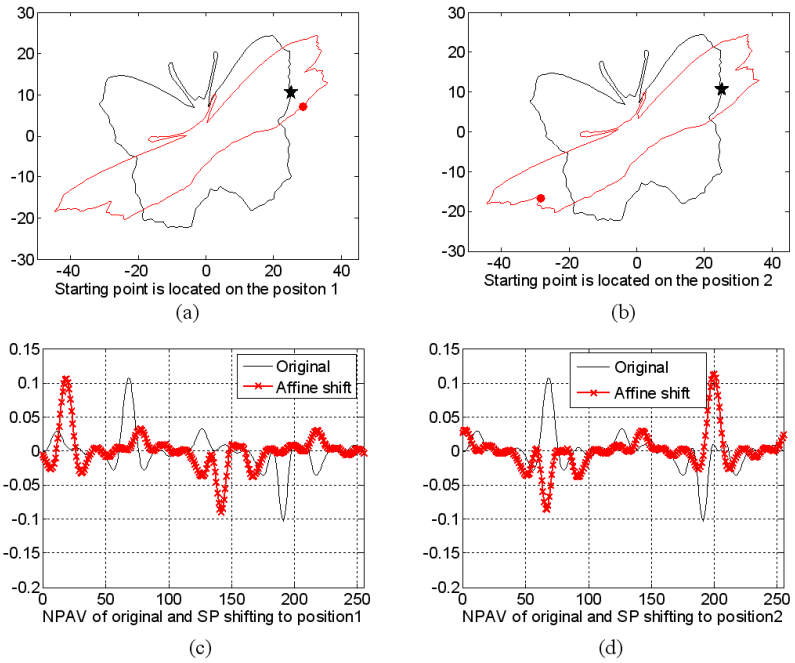
Table 3.1: Correlation under different normalized number of points

This experiment shows that under the different number of points normalized by IAN, the NPAV changes slightly under affine transforms and the greater the number of normalized points on a contour are, the lesser they are affected by affine transforms.

3.5.2.2 NPAV and position of starting point

In this section, we investigate the correlation between the different starting positions with certain affine transform. Suppose the position of the starting point (SP) is located on different positions

'1', '2', see Figure 3.19(a) and (b). These positions are respectively located at 12.5% and 37.5% from the original starting with 100% corresponding to the total number of points. Suppose the contour is normalized to 256 points. Figure 3.19(c) and (d) show the effect of different starting points to the NPAV of the original and affine contours.



(a) and (b) are the original and the affine contours with the starting point located at position '1' and '2', respectively. The star (★) is the position of the starting point on the original contour; the dot (•) is the position of the starting point on the affine transformed contour. (c) and (d) are the NPAVs of the two contours with different positions for the starting point in (a) and (b), respectively.

Figure 3.19: Illustration of the robustness of a NPAV with various positions of the starting point

As obvious from Figure 3.19, the NPAV of various starting point positions are topologically identical except for a 'circular' delay. In this way, a shift in the starting point is equivalent to a circular delay in the NPAV.

We further calculate the statistical results. To calculate the correlation between the NPAV of the original contour and that of its affine transforms with a shift of starting point. For all the shapes in MPEG-7 CE-shape-1 database, the average correlations of the various starting point positions under the same number of points normalized by IAN are presented in the Table 3.2.

Starting point shift	Average correlation coefficient
12.5%	0.989
37.5%	0.974

Table 3.2: Correlation under different positions of starting point

As it can be seen in Table 3.2, the position of the starting point on the contour does not affect the robustness of the NPAV.

3.5.2.3 NPAV versus scale and rotation transforms

Scale and rotation transforms are the important intuitive correspondences for a variety of shapes. Unsensitivity to these transforms must be regarded as a necessary condition that every shape descriptor should satisfy. In order to study the retrieval performance in term of the scale change and rotation of images, we use the test-sets Part A that is defined in CE-Shape-1 database during the standardization process of MPEG-7 [98]. The test-sets Part A includes the following two parts test. The result of test-sets Part A is the average of the two parts test results.

To test robustness to scaling, i.e., Part A-1, we create a database in which there are 70 basic shapes coming from the 70 different classes and 5 derived shapes from each basic shape by scaling digital images with factors 2, 0.3, 0.25, 0.2, and 0.1. Thus in the database, there are 420 shapes. Each of the 420 images was used as a query image. A number of correct matches were computed in the top 6 retrieved images. Thus, the best possible result is 2520 matches.

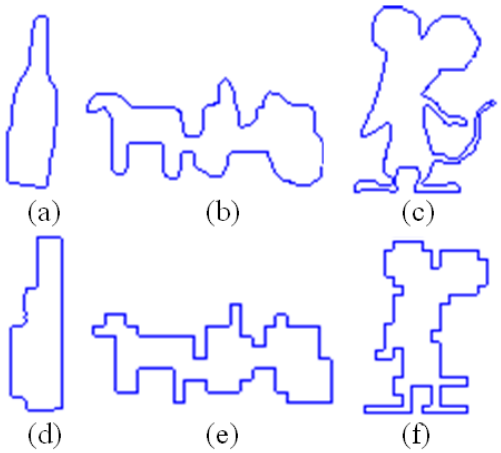
To test robustness to rotation, i.e., Part A-2, we create a database including 420 shapes also. The 70 basic shapes are the same as in Part A-1 and 5 derived shapes from each basic shape by rotation with angles: 9, 36, 45, 90 and 150 degrees. As same as Part A-1, each of the 420 images was used as a query image. The best result is 2520 matches.

The similarity rate in each experiment is calculated by taking the ratio of correct matches over the maximum number of possible matches. Table 3.3 indicates the similarity rate of comparison of NPAV descriptor with the reported results of the contour descriptors Tangent space [43], Curvature scale space (CSS) [105] and Wavelet [80].

It's clear from Table 3.3 that NPAV descriptor is comparable to other contour descriptors on this database.

Data Set	Tangent space [43]	CSS [105]	Wavelet [80]	**NPAV**
Part A1	88.65	89.76	88.04	**88.65**
Part A2	100	99.37	97.46	**99.92**
Part A	94.33	94.57	92.75	**94.29**

Table 3.3: Comparison of the retrieval results of different methods on MPEG-7 CE-Shape-1 Part A test



(a)-(c) are original contours; (d)-(f) are the scale version with factor 0.1 of (a)-(c), respectively.

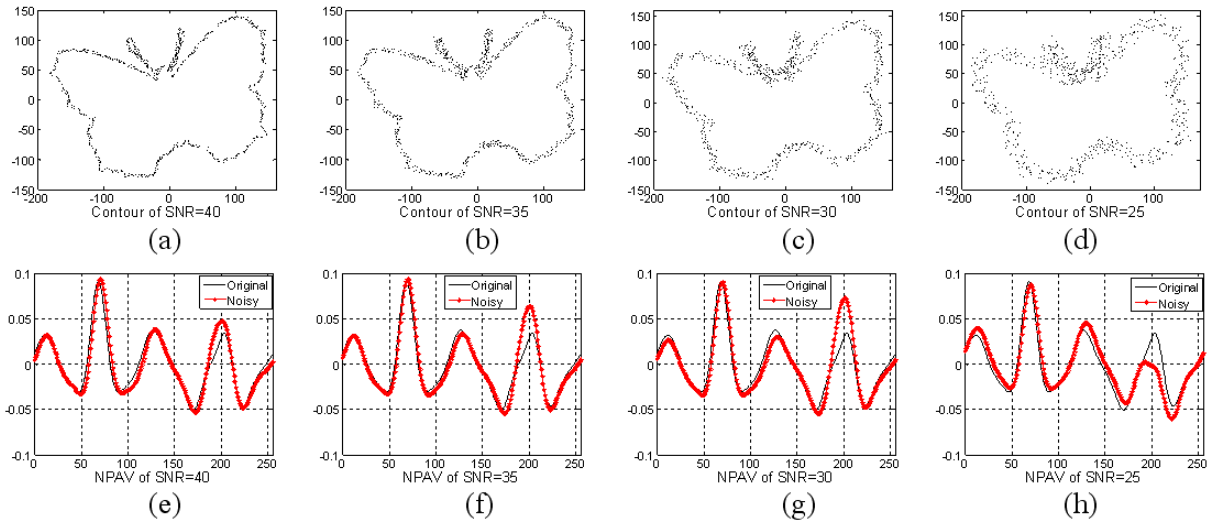
Figure 3.20: Illustration of scale vision

Some shapes obtained by scaling with the factor 0.1 are too small, Figure 3.20 presents the original object and its scaled version with factor 0.1. The scaled contours change a lot comparing to their original contours, therefore, it is hard to obtain quite correct matches on these objects.

3.5.2.4 NPAV and Noise

When we resample the edge of an object, the original shape is affected by noise and defects, which are caused by fluctuations on the boundary. The obtained shape is noisy. To simulate a more realistic noise in the context of pattern recognition, we propose a dedicated algorithm resumed in Appendix B.

To reduce the effect of noise, the curve is first smoothed. In this study the standard deviation of the filter is set to σ=2 in accordance with our experimentations. The NPAVs and their shape contaminated by the random uniform noise with different SNR are presented in Figure 3.21.



(a)-(d) are the contours contaminated by different noise power; (e)-(h) are the NPAVs of contours in (a)-(d), respectively.

Figure 3.21: Demonstration of NPAV under the condition of different SNR

This study reveals that the NPAV is robust to boundary noise and irregularities, even in the presence of severe noise. It is clear that, as the noise amplitude increases, the contours become more and more fuzzy. In order to calculate the average correlation coefficient, we do the experiments by contaminating the test contours with random uniform noise ranging from high to low SNR affecting the database. Table 3.4 shows the average correlation coefficient of all the NPAVs of shapes in the database under different SNR.

SNR	Average correlation coefficient
40dB	0.964
35dB	0.963
30dB	0.949
25dB	0.898

Table 3.4: Correlation under different SNR

It is clear from the results shown in Table 3.4 that the performances stay remarkable when the shapes are severely contaminated by noise. This shows that NPAV is suitable for use in noisy conditions.

3.5.2.5 NPAV and affine transforms

The experiment used real airplane images to test the discrimination power of the proposed affine invariance descriptor. The experiment uses the same set of twenty model plane images used in [16]. The silhouettes of these model plane images were subject to random affine transform to produce the test images. The model plane images and the test images are illustrated in Figure 3.22.

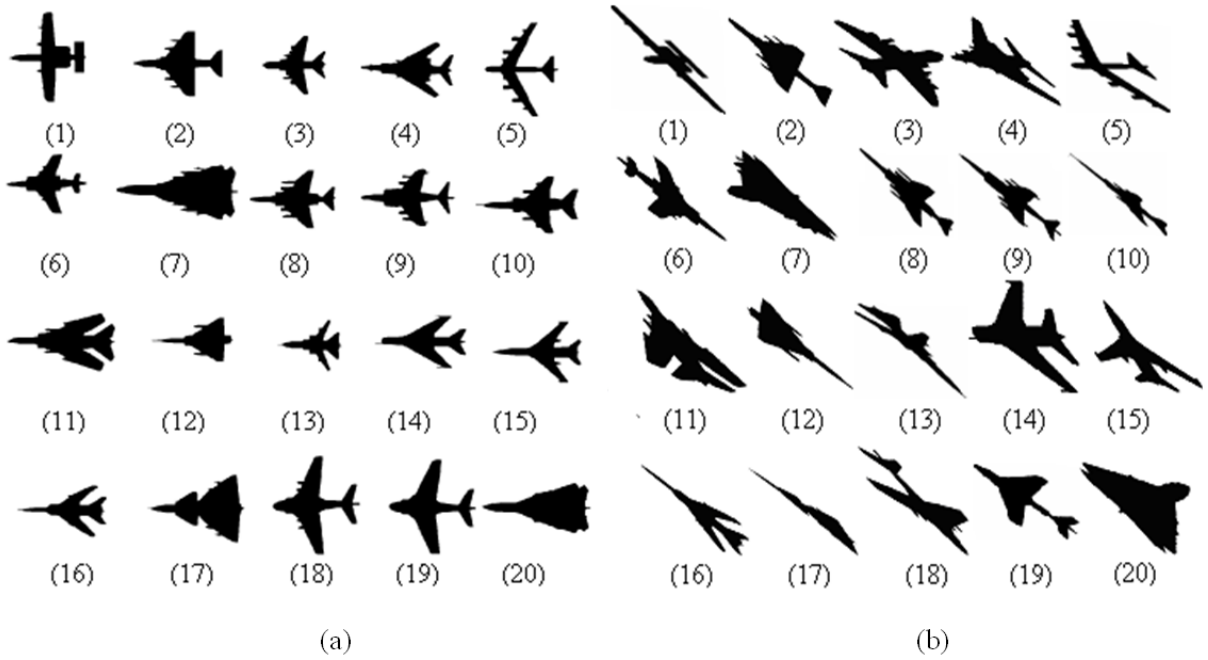


Figure 3.22: The database of plane models (a) and their corresponding affine transform images (b).

The NPAV calculated from the boundary of each test image is compared with NPAV calculated from the models using the correlation function defined in Eq. 3.13. Table 3.5 gives the best four matches for each test image. The best four matches are listed in four columns, where the best match is listed in the first column in bold. For each test, the value of the correlation function between this test image and a model (the model number is the number between parentheses) is listed. The results show that the test images are identified correctly.

By analyzing the experimental results, one can notice that NPAV is quite robust with respect to scaling, to orientation, to affine transforms, to the location of starting point and to noise. Therefore, NPAV can be used to characterize a pattern for recognition purposes.

	1	2	3	4
(1)	**0.985 (1)**	0.810 (9)	0.755 (19)	0.707 (18)
(2)	**0.922 (2)**	0.919 (8)	0.730 (6)	0.724 (10)
(3)	**0.987 (3)**	0.764 (2)	0.716 (5)	0.516 (8)
(4)	**0.990 (4)**	0.808 (11)	0.706 (20)	0.705 (10)
(5)	**0.952 (5)**	0.688 (3)	0.634 (8)	0.625 (2)
(6)	**0.987 (6)**	0.833 (10)	0.773 (13)	0.723 (18)
(7)	**0.982 (7)**	0.695 (13)	0.646 (14)	0.639 (17)
(8)	**0.972 (8)**	0.786 (2)	0.659 (5)	0.638 (16)
(9)	**0.987 (9)**	0.886 (10)	0.792 (1)	0.749 (19)
(10)	**0.925 (10)**	0.823 (4)	0.772 (19)	0.771 (6)
(11)	**0.967 (11)**	0.797 (4)	0.718 (20)	0.606 (15)
(12)	**0.984 (12)**	0.570 (6)	0.570 (7)	0.569 (17)
(13)	**0.907 (13)**	0.668 (6)	0.637 (7)	0.632 (1)
(14)	**0.982 (14)**	0.812 (15)	0.622 (16)	0.613 (7)
(15)	**0.969 (15)**	0.800 (16)	0.746 (11)	0.701 (20)
(16)	**0.962 (16)**	0.764 (20)	0.714 (15)	0.704 (11)
(17)	**0.780 (17)**	0.760 (12)	0.621 (7)	0.585 (13)
(18)	**0.990 (18)**	0.970 (19)	0.859 (10)	0.759 (9)
(19)	**0.991 (19)**	0.984 (18)	0.848 (10)	0.762 (9)
(20)	**0.995 (20)**	0.770 (11)	0.696 (4)	0.659 (16)

Table 3.5: The best four NPAV matches between the test images and the model images

3.5.3 Conclusion about NPAV

A new method of extracting invariants of a shape under affine transform is proposed in this section. Our representation is based on the association of two parameters: the affine arc length and the enclosed area, viz., we normalize a contour to affine invariance by the iso-area. We then prove two theorems and a deduction. They reveal that, for a filtered contour, the partial enclosed area is linear under affine transforms. We further define the affine-invariance vector: the normalized partial area vector (NPAV). A number of experiments applied to the MPEG-7 CE-shape-1 database and real images database demonstrate that NPAV is quite robust with respect to affine transforms and noise, even in the presence of severe noise.

3.6 Conclusion

In this chapter, we have made a comprehensive study of affine invariant shape descriptions based on iso-area normalization (IAN). First we introduced the representation of affine transformation and some affine invariant parameters about shape contours. And then an approach of IAN of a contour has been presented in details. Finally, two affine invariant shape descriptors based on IAN are proposed: scale-controlled area-difference shape descriptor (SCAD) and normalized partial area vector (NPAV).

With inspiration from psychophysics where the human vision distinguishes the object only from characteristics of main segments, we proposed the first descriptor SCAD. This proposed descriptor has three major contributions: 1) it compares the same segment under different scales

representation; 2) it chooses the appropriate scales by applying a threshold to the shape shortest-segment; 3) it proposes the algorithm and the conditions of merging and removing the short segments. This proposed method is robust under affine transformations with low computational complexity.

We have defined the second affine invariant descriptor, NPAV. We then prove and demonstrate, by two proposed theorems, that for any linearly filtered contour, the area of a triangle, whose vertices respectively are the centroid of the contour and a pair of successive points on the normalized contour, remains linear under affine transforms. So the relation between a filtering and affine transformation is set up. In theory, NPAV remains absolutely invariant under affine transformations. Experimental results, executed on the MPEG-7 CE-Shape-1 and twenty plane model images commonly used in the literature, indicate that the proposed method is unaffected by boundary starting point variations and affine transforms even in the case of high deformations and serious noise contamination on the shapes.

In the next chapter, a shape-based image description and similarity measure approach will be proposed for similarity-based image retrieval.

Chapter 4

Chord context descriptor

In this chapter, we propose a new effective shape descriptor, chord context, for shape description in content-based image retrieval. For a given shape, the chord context describes a frequency distribution of chord lengths with different orientations. The basic idea of chord context is to calculate the lengths of all parallel chords with the same interval in a shape, and to build a histogram of their lengths in each direction. The sequence of extracted feature vectors forms the feature matrix for a shape descriptor. Because all the viewpoint directions, considered with a certain angle interval, are chosen to produce the chord length histogram, this representation is unlike conventional shape representation schemes, where a shape descriptor has to correspond to key points such as maxima of curvature or inflection points, for example, Smooth Curve Decomposition [5], Convex Hull [106] and Curvature Scale Space (CSS) [45, 17], etc. The proposed method needs no special landmarks or key points. There is also no need for certain axes of a shape. The proposed descriptor scheme is able to capture the internal details, specifically holes, in addition to capturing the external boundary details.

A similarity measure is defined over chord context according to their characteristics and it confirms efficiency for shape retrieval from a database. The proposed method of shape descriptor and retrieval is shown to be invariant under image transformations, rotations, scaling and to be robust to non-rigid deformations, occultation and boundary perturbations by noise. In addition, the size of the descriptor attribute is not very great; it has low-computational complexity compared to other similar methods.

Several experiments have been conducted on the MPEG-7 CE-1 database [16], COIL-100 database and Kimia silhouettes [67, 107]. The results demonstrate the feasibility of the chord context descriptor methodology and also highlight its advantages over other existing methodologies. For more detail information, please refer to our paper [108].

The remainder of this chapter is organized as follows. First the principle of chord context and similarity measure of chord context are presented. Then we demonstrate the experimental results obtained in our approach, as well as the comparisons with other existing approaches. Finally, this chapter conclusion is formulated and presented.

4.1 Chord context

This section details the proposed method, chord context, for extracting attributes from the contour or silhouette of a shape. It then proposes a method of measuring similarities between two shapes.

4.1.1 Feature extraction

Chord context analysis corresponds to finding the distribution of all chord lengths in different directions in a given shape. So for each direction, we observe all the chords in the shape. Figure 4.1 shows an example of chords in direction θ. The bold lines are the chords of the shape.

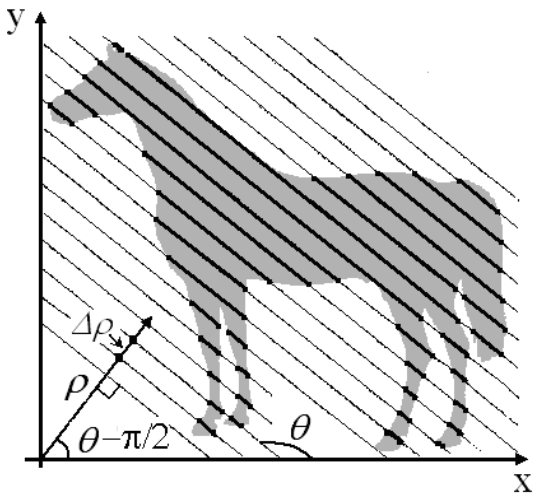


Figure 4.1: Representation of chords in direction θ with the interval $\Delta\rho$

A set of lines $T(\rho, \theta)$ is defined by

$$\rho = x\cos(\theta - \pi/2) + y\sin(\theta - \pi/2),\ \theta \in [0,\ \pi]\ and\ \rho \in (-\infty, +\infty)$$

The chords are defined by the parts of these lines within the domain of the binary shape (see Eq. 1.1). So a shape can be represented by a discrete set of chords sampled from its silhouette. Considering different angles θ, the number and length of chords obtained in different directions may not be the same, except in the case of a circle. One way to capture this information is to use the distribution of chord lengths in the same direction in a spatial histogram.

Concretely, let us assume that the set of chords in directions θ_i are represented by $C = \{c_{i,n} \mid n \in [1, N_i]\}$, where N_i is the number of the chords in the direction θ_i. Let $L(c_{i,n})$ be the length of chord $c_{i,n}$. So we can compute a histogram $\mathbf{h_i}$ in direction θ_i by

$$\mathbf{h_i} = \left[h_i^1, h_i^2, \cdots, h_i^l, \cdots h_i^{L_{i,max}}\right] \tag{4.1}$$

where $h_i^l = \sharp\{L(c_{i,n}) \in bin(l)\}\ l \in [1, L_{i,max}]$, $L_{i,max}$ is the longest chord in direction θ_i.

In order to capture the details of a shape, the interval $\Delta\rho$ of ρ, i.e., the distance between two parallel chords, should not be great. In practice, $\Delta\rho = L_{max}/(50 \sim 100)$, where L_{max} is the length of the longest axis of the shape. The histogram $\mathbf{h_i}$ of Figure 4.1 in direction θ_i is shown in Figure 4.2.

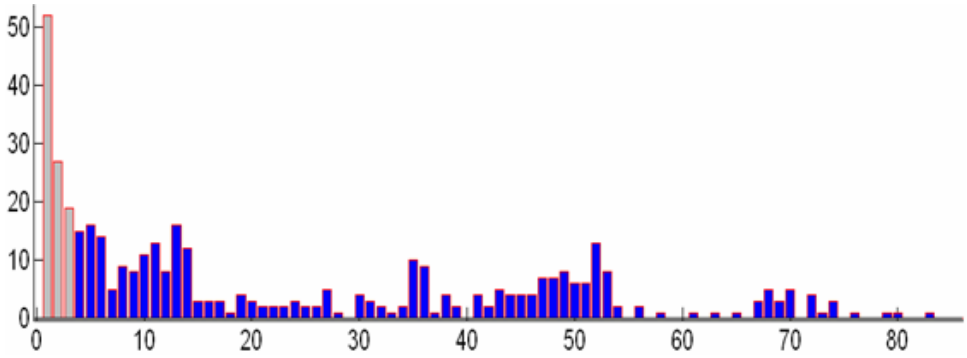


Figure 4.2: Histogram of chord lengths in direction θ_i for the shape shown in Figure 4.1

An excessive number of very short chords is counted when line T is close to a tangent along the edge of the shape (see Figure 4.3). This is because a scraggy edge is produced by the minor disturbances resulting from digitization noise or normalization of the image to a certain size. In fact, these uncertain short chords are harmful to our shape descriptor, so we remove these very short chords directly. Empirical tests show that, if the largest size of the shape is 128 pixels, then we can consider the set of chords whose length is shorter than 4 to be too short. So they should be discarded. In Figure 4.2, the first 3 bins should be removed.

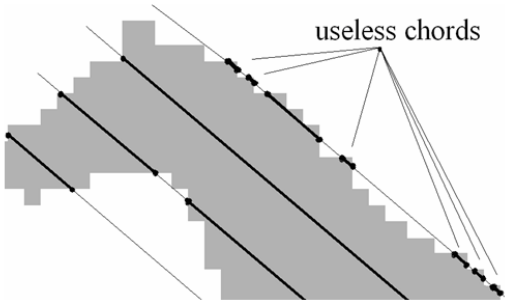


Figure 4.3: Illustration of producing very short chords

With θ increasing from 0 to 179 degrees, all the chords in different directions in the silhouette can be recorded. If we divide the orientation range $[0, 179]$ into D', then we can obtain D' histograms $\mathbf{h_i}$, $i \in [0, D' - 1]$. They form a matrix M arranged by a set of histograms with column vector $\mathbf{h_i}$ according to the order of angles:

$$M = [\mathbf{h_0}, \mathbf{h_1}, \cdots, \mathbf{h_{D'-1}}]$$

The matrix M extracted from the shape in Figure 4.1 is shown in Figure 4.4. The abscissa is the direction (here each value covers 2 degrees); the y-axis is the length of chords. A matrix element is the number of equal length chords whose direction and length are given by the value of abscissa and y-axis, respectively.

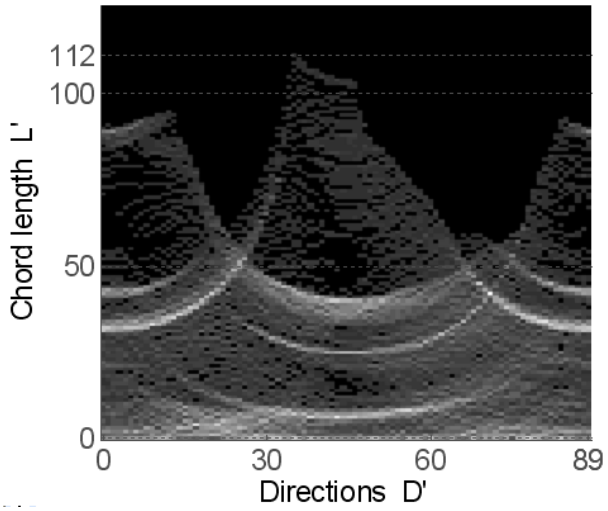


Figure 4.4: The combination of the chord length histograms of the shape in Figure 4.1 with all orientations

The value in each row of the matrix M is the number of the chords with same length in different directions; and each column is the chord length histogram in the same direction.

Directly using this matrix as shape descriptor is not very attractive: it will be too large. For example, in an image with 128×128 pixels, the longest possible chord in the shape is $128 \times \sqrt{2} \approx 181$. So, if $D' = 90$, the size of the matrix will be $181 \times 90 = 16290$. Clearly, it is not appropriate as a direct feature of a shape. In order to reduce the size of the matrix M, at the same time, make the extracted feature invariant with scale transforms, we normalize this matrix M as follows:

1. Find the maximum of non-zero bin L' for all the histograms in the matrix M. In Figure 4.4, for example, $L'_{max} = 112$. Then remove all the bins that are greater than L'_{max}, and form a matrix M' with dimension $L'_{max} \times D'$:

$$M' = \left[\mathbf{h'_0}, \mathbf{h'_1}, \cdots, \mathbf{h'_{D'-1}}\right]$$

2. For the next normalization, we expand the matrix M' to matrix M'', using a wrap-around effect:

$$M'' = \left[\mathbf{h'_{D'-2}}, \mathbf{h'_{D'-1}}, \mathbf{h'_0}, \mathbf{h'_1}, \cdots \mathbf{h'_{D'-1}}, \mathbf{h'_0}, \mathbf{h'_1}\right]$$

3. The matrix M'' is subsampled by the bicubic algorithm interpolation down to a new matrix F with the dimension $L \times D$. For convenience, D is even. The bicubic algorithm interpolation means that the interpolated surface is continuous everywhere and also continuous in the first derivative for all directions. Thus, the rate of change in the value is continuous. The feature matrix F can be represented by:

$$F = [\mathbf{f_0}, \mathbf{f_1}, \cdots, \mathbf{f_{D-1}}]$$

where $\mathbf{f_i}$, $i \in [0, D-1]$, is a L dimensions column vector given by $\mathbf{f_i} = \left[f_i^1, f_i^2, \cdots, f_i^L\right]^T$.

The feature matrix F is the attribute of the shape; we call it "chord context descriptor".

If $L = 30$ and $D = 36$, the chord context derived from Figure 4.1 is shown in Figure 4.5. Each column is the normalized histogram in a given direction.

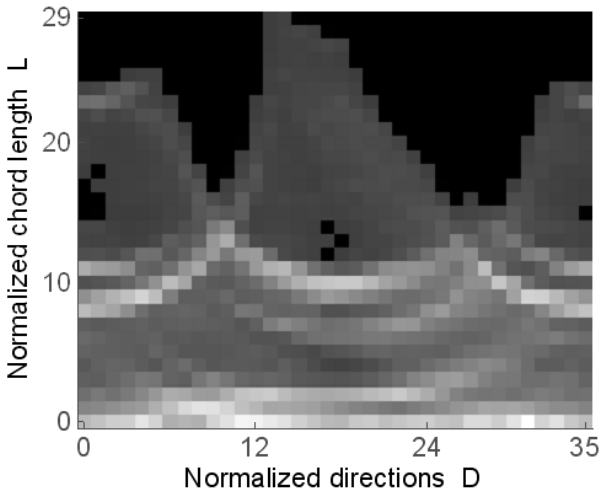


Figure 4.5: Chord context of Figure 4.1 with 30 rows and 36 columns

The experiment in section 4.2 shows that chord context as the feature of a shape can retain the visual invariance to some extent.

4.1.2 Evaluation of chord context descriptor

In determining the correspondence between shapes, we aim to meet the distance between two feature matrices. In this study we present an approach directly with *Character Matrix Distance (CMD)* and then we improve it by introducing the *Perpendicular Chord Length Eccentricity (PCLE)*.

4.1.2.1 Character Matrix Distance

In the first phase, we calculate all the distances between the query feature matrix and the model feature matrix while circularly shifting its histograms one by one. Similar shapes have similar histograms in a same direction, and the arrangement order of these histograms is also similar . To calculate the distance between two attributes of shapes, we first calculate the distance between each corresponding histogram, according to their arrangement orders, and then calculate the sum of all these distance values. Regarding rotation invariance, we shift the model feature matrix by one histogram, i.e. change the direction used to obtain the histograms, and repeat the same step to calculate the sum of all the values of these distances between the two feature matrices.

We assume that the query feature matrix is F_Q and the model feature matrix is F_M. F_Q and F_M are given by

$$F_Q = [\mathbf{fq_0}, \mathbf{fq_1}, \cdots, \mathbf{fq_{D-1}}] \; and \; F_M = [\mathbf{fm_0}, \mathbf{fm_1}, \cdots, \mathbf{fm_{D-1}}]$$

According to subsection 4.1.1, $\mathbf{f}\alpha_\mathbf{i}$, where α is q or m, $i \in [0, D-1]$, is an L dimensions column vector

$$\mathbf{f}\alpha_\mathbf{i} = \left[f\alpha_i^1, f\alpha_i^2, \cdots, f\alpha_i^L\right]^T$$

So the set of similarity distance is given by

$$DistF_{F_Q,F_M}(n) = \sum_{i=0}^{D-1} DistH(\mathbf{fq_i}, \mathbf{fm_j}),\ j = mod(i+n, D) \tag{4.2}$$

where $n \in [0, D-1]$ is the number of shifts applied to each histogram in the model feature matrix. The formula shows that the set of similarity distance is the sum of the distance *DistH* between two corresponding normalized histograms in two feature matrices.

To quantify the similarity between two histograms, as introduced in Chapter 1, there are many methods being reported: Minkowski-form, Kullback-Leibler Divergence, Jeffrey Divergence, Quadratic-form, χ^2 statistics, etc. Due to the properties of the chord context histograms:

- they have the same number of bins;
- the value in each bin has great variances; some of values are even zeros, cf. Figure 4.2.

We compare χ^2 statistics distance

$$DistH_{\chi^2}(\mathbf{fq_i}, \mathbf{fm_j}) = \frac{1}{2L} \sum_{k=1}^{L} \frac{\left(fq_i^k - fm_j^k\right)^2}{\left(fq_i^k + fm_j^k\right)}$$

and our proposed distance formula defined here by

$$DistH(\mathbf{fq_i}, \mathbf{fm_j}) = \begin{cases} 0, & fq_i^k = 0\ and\ fm_i^k = 0,\ k \in [1,\ L] \\ \frac{1}{L}\sum_{k=1}^{L} \frac{|fq_i^k - fm_j^k|}{\max\left(fq_i^k, fm_j^k\right)}, & otherwise \end{cases}$$

on the database of Kimia silhouettes with 216 shapes [67]. This database is shown in Appendix C.1. For convenience, we consider the minimum value of the distance set as the similarity distance and call it *Character Matrix Distance (CMD)*.

$$CMD_{F_Q,F_M} = \min_{n=0}^{D-1} DistF_{F_Q,F_M}(n) \tag{4.3}$$

The comparison result of precision versus recall is shown in Figure 4.6 (cf. Subsection 1.4.1).

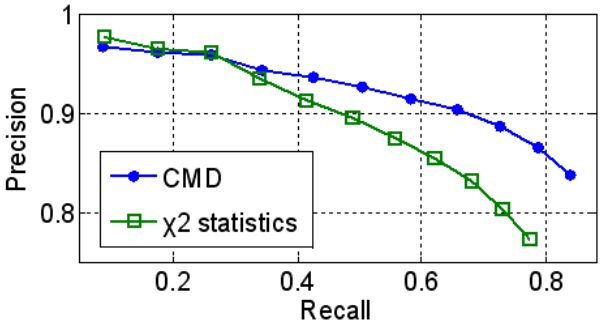


Figure 4.6: The precision-recall diagrams for indexing into the database of Kimia silhouettes with 216 shapes

It is clear from Figure 4.6, that our proposed distance formula is better than χ^2 statistics on this similarity measure.

4.1.2.2 Perpendicular chord length eccentricity

In order to improve the accuracy of the similarity measure, we propose a new concept, *Perpendicular Chord Length Eccentricity (PCLE)* as following:

$$PCLE(i) = \begin{cases} \left\|\mathbf{f_i} - \mathbf{f_{i+(D/2)}}\right\| & i \in \left[0, \frac{D}{2} - 1\right] \\ \left\|\mathbf{f_i} - \mathbf{f_{i-(D/2)}}\right\| & i \in \left[\frac{D}{2}, D - 1\right] \end{cases} \tag{4.4}$$

where $\|\bullet\|$ is the norm; $\mathbf{f_i}$, $i \in [0, D-1]$ (D is even) are vectors of a feature matrix. They are also normalized histograms of chord length in a shape. So Eq. 4.4 is the Euclidean distance between any two chord length histograms of perpendicular direction. As the symmetric norm, we have

$$PCLE(i + \frac{D}{2}) = PCLE(i),\ i \in [0, \frac{D}{2} - 1]$$

Clearly, *PCLE* represents the perpendicular direction chord feature in a shape. To compare the query's *PCLE* P_Q and the model's *PCLE* P_M, we define the distance between P_Q and P_M as follows:

$$D\_PCLE_{P_Q,P_M}(n) = \sum_{i=0}^{D-1} \frac{|P_Q(i) - P_M(j)|}{max(P_Q(i), P_M(j))},\ j = mod(i+n, D)$$

where n is the number of shifts applied to each histogram in the model *PCLE* as in Eq. 4.2.

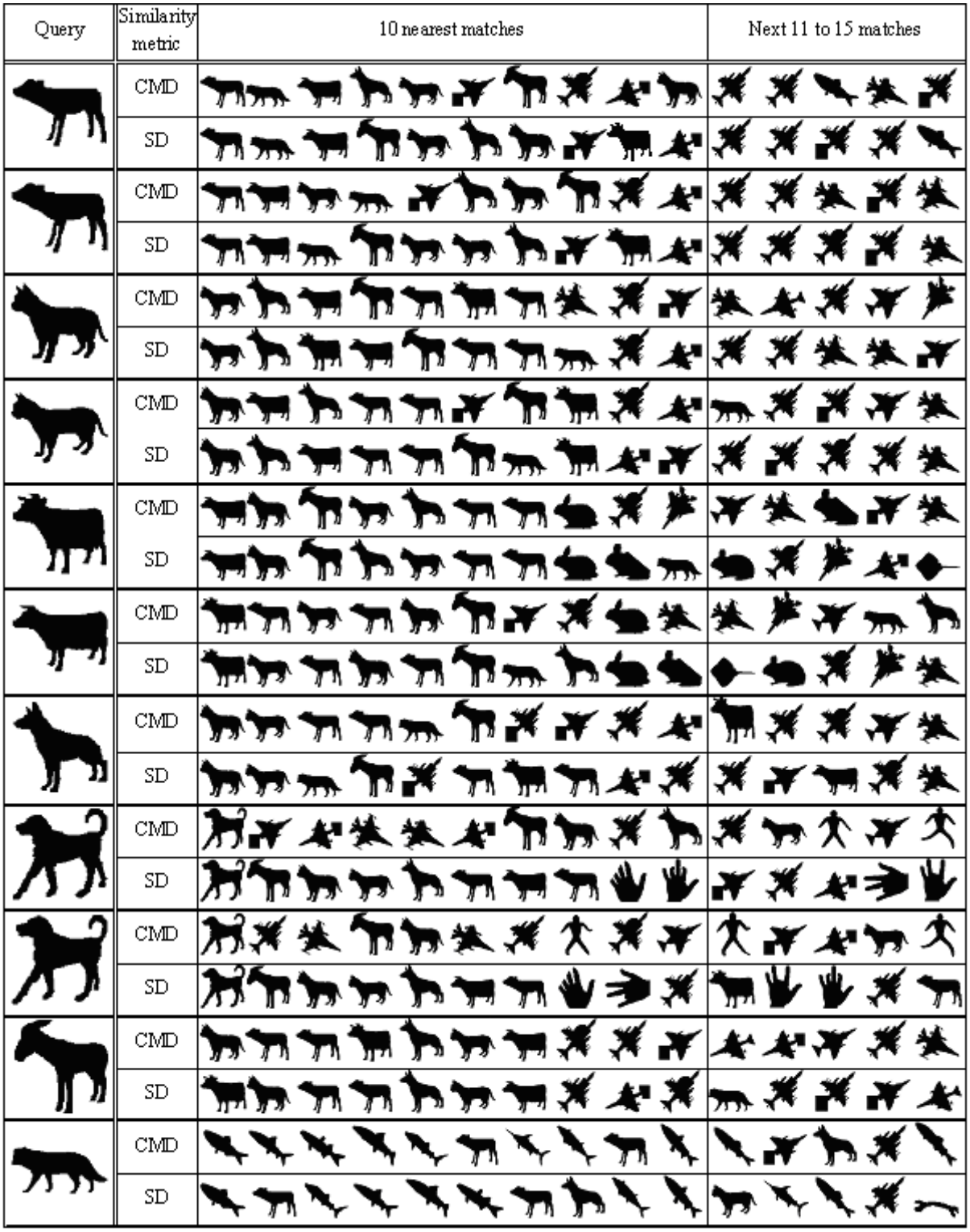
Intuitively, we assume that, in general, if shape S_1 is more similar to shape S_2 than shape S_3, then the smallest value of $D\_PCLE_{P_{S_1},P_{S_2}}$ is less than the smallest value of $D\_PCLE_{P_{S_1},P_{S_3}}$.

So we can use D_PCLE to adjust the similarity distance of the *Character Matrix Distance (CMD)* to improve retrieval precision and recall. The combined similarity metric of shape query and shape model is computed using a weighted sum:

$$SD_{Q,M} = \alpha \cdot CMD_{F_Q,F_M} + (1-\alpha) \min_{n=0}^{D-1}(D\_PCLE_{P_Q,P_M}(n)) \tag{4.5}$$

where $\alpha \in [0, 1]$.

Let us explain this hypothesis by the following experiments, where we have used $\alpha = 0.85$. First let us look at the experiment running on the Kimia silhouettes set of 99 shapes [67]. This database is shown in Appendix C.2. Figure 4.7 shows the two retrieval results from querying the 'Quadruped' category. One shows the retrieval results only with the similarity distance of *Character Matrix Distance (CMD)* (Eq. 4.3) and the other with the combined similarity metric weighted sum *SD* (Eq. 4.5). Each shape is matched against all the other shapes in the database. As there are 11 shapes in the 'Quadruped' category, up to 10 nearest neighbors can be retrieved from the same category. We count in the n^{th} (n from 1 to 15) nearest neighbors the number of times that the test image is correctly classified. The best possible result is 110 matches (except the query itself) in all 10 nearest matches. With the similarity metric *CMD*, we found 63 matches in the first ten retrieved shapes, recall is 63/110=57.3%, and there were 70 matches in the first fifteen retrieved shapes. Whereas with the similarity metric *SD*, we found 80 matches in the first ten retrieved shapes, recall is 80/110=72.7%, and there were 85 matches in the first fifteen retrieved shapes. The result shows that the recall rate in the first ten retrieved shapes was improved by 15.4 percentage points for the 'Quadruped' cluster when we used the similarity metric *SD* instead of *CMD*. It also shows a good performance rate when compared with [109], [59] and [85] which has the same retrieval results of 51 matches in the first ten retrieved shapes.



The 11 quadruped shape queries in the dataset are shown in the first column. The 10 nearest retrieved shapes for each query are shown in order (from small similarity distance to large similarity distance) in the 3^{rd} column by their similarity metric CMD and SD. The next five matches are shown in the 4^{th} column for completeness.

Figure 4.7: Illustration of retrieval results from the 'Quadruped' category in Kimia silhouettes set of 99 shapes

Let us look at the statistical results. We compare the retrieval result using *CMD* to that using similarity metric *SD* by calculating precision versus recall in the Kimia silhouette datasets of 99 and 216 shapes [67]. The results are shown in Figure 4.8.

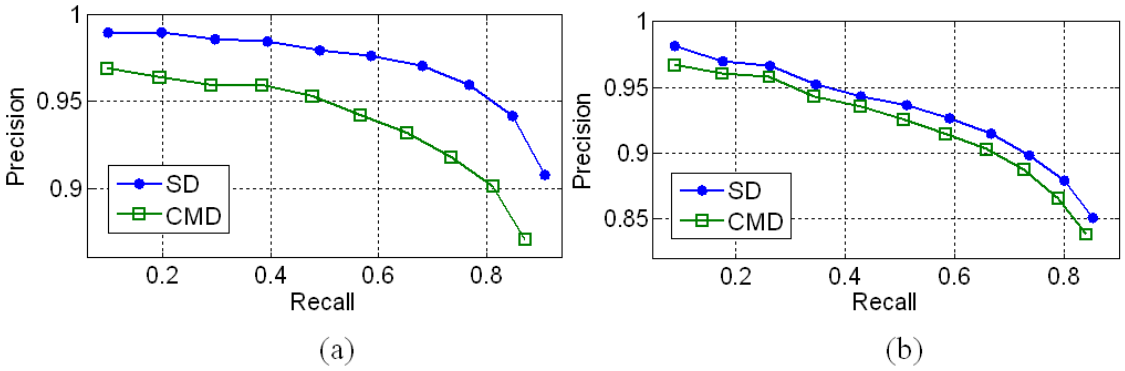


Figure 4.8: The precision-recall diagrams for indexing into the database of Kimia silhouettes with (a) 99 shapes and (b) 216 shapes

The experimental results show significant superiority of *SD* compared to *CMD* in retrieval process.

4.2 Experimental evaluations of chord context matching

In this section, we present the results obtained during a shape recognition process to study the comparative performances of the proposed algorithm. We show that the chord context matching is effective in the presence of commonly occurring visual transformations like scale changes, boundary perturbations, viewpoint variation, non-rigid transform and partial occultation. We also compare its results with ten other well-known algorithms. All the experiments are conducted on the standard database: MPEG-7 CE-shape-1 database (1400 shapes) [2], Columbia University Image Library Coil-100 database (7200 images) [110], and 3 databases of Kimia silhouettes [67]. In all the experiments, the feature matrix was normalized to 30 bins and 36 directions; the similarity measure uses Eq. 4.5 with $\alpha = 0.85$.

4.2.1 Scale and rotation transforms

In order to study retrieval performance in terms of scale changes and image rotations, we use the test-sets Part A introduced in Chapter 3. One can recall that the MPEG-7 CE-Shape-1 database consisting of 1,400 shapes semantically classified into 70 classes.

The similarity rate in each experiment was calculated by taking the ratio of correct matches to the maximum number of the relevants to rotation and scaling. Table 4.1 indicates the similarity rate of comparison by the chord context descriptor with the reported results of certain studies: Tangent space [43], Curvature scale space (CSS) [105], Zernike moments[111], Wavelet [80] and Beam angle statistics (BAS) [60]. Note that the proposed descriptor has the best performance in all the experiments.

Data Set	Tangent space [43]	CSS [105]	Zernike moments[111]	Wavelet [80]	BAS [60]	**Chord context**
Part A1	88.65	89.76	92.54	88.04	90.87	**99.37**
Part A2	100	99.37	99.60	97.46	100	**100**
Part A	94.33	94.57	96.07	92.75	95.44	**99.69**

Table 4.1: Comparison of the retrieval results of different methods on MPEG-7 CE-Shape-1 Part A test

The results show that chord context is very resistent to scale and rotation transforms. Reviewing the extracted attribute algorithm in section 4.1, we are not surprised by the almost perfect results. The attribute matrix is obtained by the statistic of all the chord lengths of a shape in all directions. The rotation of the shape affects the attribute matrix only when shifting the chord length histograms. In the similarity measure, we have considered this point and compared the two attribute matrices by shifting the histograms of either matrix. The rotation of plane shapes does not affect the retrieval result. Since we normalize all the images to a certain size before extracting their features, the scale transform of a shape does not significantly affect the retrieval result.

4.2.2 Boundary perturbations by noise

The query shape can be perturbed by different noises. This may simply result from digitization. As a reminder, to fight perturbations resulting from shape digitization, and in order to alleviate the influence of boundary perturbation, we have removed the very short chords in attribute matrices. To evaluate the performance of chord context when boundary perturbations are present, we use noisy images with different noise powers as queries to retrieve the relevant image in a database. We generate a 20 sub-database test-set based on MPEG-7 CE-Shape-1. In each sub-database, there are 70 shapes from 70 different classes according to their orders in the database, i.e., all the shapes belong to different classes in each sub-database. The query shapes are all 70 shapes in each sub-database subjected to noise with 4 different noise powers. Thus, the best possible result in each sub-database is 70 matches for each noise power. To simulate the boundary perturbations by noise, we shift the positions of certain boundary points selected randomly. In this experiment, 50% boundary points are shifted on a shape. The amplitude of this shift is controlled by a uniform random value. Suppose the average distance of all the points on the edge of a shape to its centroid is D. We then define the signal-to-noise ratio (SNR) as follow:

$$SNR = 20 \log \frac{D}{r} (dB)$$

where r is the largest deviation of the points on the edge. Figure 4.9 shows an example of an original shape and its contaminated shapes produced by random uniform noise with SNR equal to 30dB, 25dB, 20dB and 15dB.

From left to right, the original and the resulting noised shapes at an *SNR* of 30dB, 25dB, 20dB and 15dB.

Figure 4.9: Examples of noisy shapes from a model in MPEG-7 CE-Shape-1

Table 4.2 shows the average similarity rates of the 20 Sub-Databases at 4 *SNR*s.

SNR(dB)	30	25	20	15
Average Similarity Rate (%)	99.9	99.8	99.1	82.9

Table 4.2: Average similarity rates of the 20 sub-databases at SNR of 30dB, 25dB, 20dB and 15dB

From the results in Table 4.2, we notice that using noise to disturb boundaries does not produce significant differences between similar shapes. As the chord context descriptor utilizes the edge as well as the region feature of a shape, it can bear boundary disturbances due to noise to certain extent.

4.2.3 Partial occultation

In general, a global descriptor is not robust when a shape is partially occulted. Since the chord context descriptor has the statistical information for a shape, this drawback is alleviated to some extent. To evaluate robustness to partial occultation we ran experiments using the same sub-databases as mentioned in Subsection 4.2.2. We occulted the shapes, applying 4 different percentages of occultation from the left, Figure 4.10(a), or right, Figure 4.10(b), respectively to them, in raster-scan order. The occultation percentages were 5%, 10%, 15% and 20%. Each occulted shape is retrieved in its sub-database as a query. Thus, the best result in each sub-database is 70 matches for one occultation.

(a) (b)

(a) Left side occulted objects; they are occulted by 5%, 10%, 15% and 20%;
(b) Right side occulted objects; they are occulted by 5%, 10%, 15% and 20%.

Figure 4.10: Examples of occulted shapes from a model in MPEG-7 CE-Shape-1

Table 4.3 shows the average similarity rate of the 20 sub-databases on the 4 partial examples of occultation.

Occultation (%)		5	10	15	20
Average Similarity Rate (%)	left	99.5	91.3	78.6	58.7
	Right	98.9	94.1	80.7	64.6
	Average	99.2	92.7	79.7	61.7

Table 4.3: Average similarity rates of the 20 sub-databases at shape occultation of 5%, 10%, 15% and 20%

It is clear from Table 4.3, that small occultations do not significantly affect chord context. However, the problem of significant occultation remains to be explored. The results show that chord context are robust to minor occultation.

4.2.4 Similarity-based evaluation

The performance in similarity-based retrieval is perhaps the most important of all tests performed. In order to demonstrate the performance of chord context when deformed parts are present, we turn to three shape databases. All three databases were provided by Kimia's group [67, 107].

4.2.4.1 Experiment 1

The first database is Kimia silhouettes database which contains 25 images from 6 categories (cf. Appendix C.3). Each row shows instances of a different object category.

This property has been tested by shape contexts [22], Sharvit et. al [107], Gdalyahu et. al [112] and Ling et. al [113]. The retrieval results are summarized as the number of first, second, and third closest matches that fall into the correct category. The results are listed in Table 4.4. It shows that our proposed method outperforms the first 3 reported methods. For the fourth approach, chord context is slightly better than it in the Top 2 closest matches.

Methods	Top 1	Top 2	Top 3
Sharvit et. al [107]	23/25	21/25	20/25
Gdalyahu et. al [112]	25/25	21/25	19/25
Belongie et. al [22]	25/25	24/25	22/25
Ling et. al [113]	25/25	24/25	25/25
Chord context	**25/25**	**25/25**	**23/25**

Table 4.4: Comparison of the retrieval results of different methods on the Kimia silhouettes with 25 shapes

4.2.4.2 Experiment 2

The second database contains 99 images from nine categories with 11 shapes in each category (cf. Appendix C.2). It has been tested by D.S. Guru [59], Shape contexts [22], Bernier [109] and Tabbone [85]. Each shape was used as a query to which all other shapes were compared and thus 9,801 shape comparisons were made. Ideal results would be that the 10 closest matches (except the query itself) belong to the same category. The results are summarized by precision-recall diagrams in Figure 4.11. The proposed method shows better precision and recall rate than the other methods.

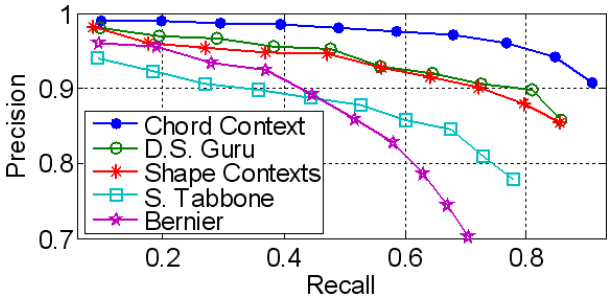


Figure 4.11: Comparison of precision and recall rates of different methods on the Kimia Data Set of 99 shapes

4.2.4.3 Experiment 3

The third database contains 216 images from 18 categories with 12 shapes in each category (cf. Appendix C.1). All the shapes were selected from the MPEG-7 test database. It has been tested by shape contexts [22]. As in the case of second database, a comparison of the results of our approach to the shape context method is given in Figure 4.12. As we see in Figure 4.12, the two precision/recall curves cross. This means that the shape contexts [22] method performs better for small answer sets, while our proposed method performs better for larger answer sets. According to [114], the method achieving higher precision and recall for large answer sets is considered to be the best, so our method is better than shape contexts in this database.

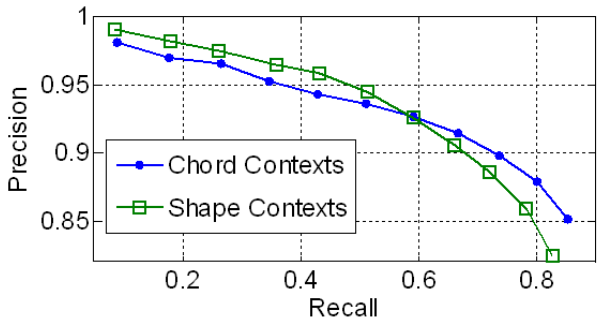


Figure 4.12: Comparison of precision and recall rates of different methods on the Kimia Data Set of 216 shapes

From the above results, we see that chord context produces outstanding performance in the presence of non-rigid deformations.

4.2.5 Viewpoint variations

For a better evaluation in the realistic context of image retrieval with industrial vision where the picture of an object from real world is converted and observed from different viewpoints, we have performed tests on shapes extracted directly from pictures. To test the retrieval performance of the proposed method in the presence of viewpoint changes, we turned to the Columbia University Image Library Coil-100 3D object dataset[1]. This dataset contains 7,200 color images of 100 household objects and toys. These 100 object are shown in Figure 4.13.

[1] http://www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php

Figure 4.13: Columbia University Image Library Coil-100 3D object

Each object was placed on a turntable with rotations in azimuth of 0 through 360 with 5 increments, resulting in 72 images per object. We have converted the color images to gray scale images, and then into shapes by using the same grey-value threshold settings for the whole set. Since the shapes are projections of 3D objects by a simple grey-value threshold, not only their silhouettes can change from one viewpoint to another; the lighting and object textures may also differ. Figure 4.14 shows some examples. The first row shows the original images and the second row shows their corresponding silhouettes as produced by grey-value threshold. In 1^{st} and 2^{nd} columns, the duck's silhouettes change significantly due to a change of viewpoint. In the 3^{rd} and 4^{th} columns, the red pepper's silhouettes change significantly due to a difference in lighting. In 5^{th} and 6^{th} columns, the tin's silhouettes change significantly due to a change in textures.

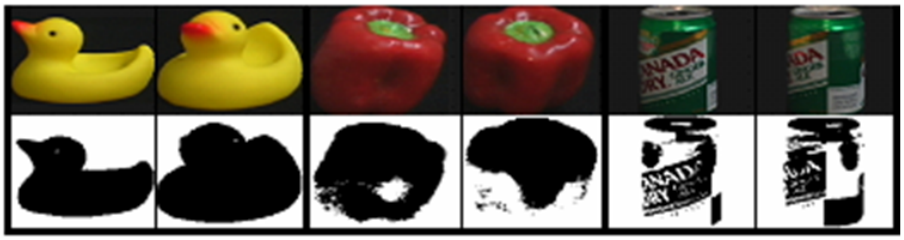


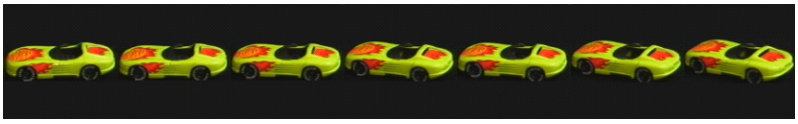
Figure 4.14: 6 images taken as an example from the COIL-100 3D object dataset

In the following subsections we present two experiments showing the performance of the proposed method on these test sets.

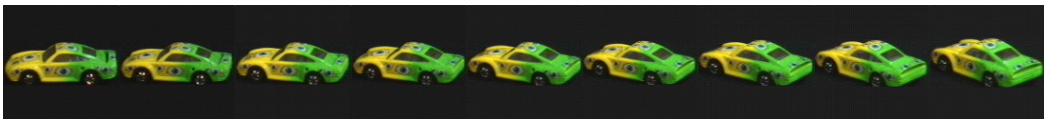
4.2.5.1 Experiment 1

First we compare our approach to the shape context [115] in Coil-100 3D object database. We have converted the color images into shapes, then selected 3 images per object with a 15° viewpoint interval. In this experiment, the 3 images of each object are the viewpoints of 0°, 15° and 30°, respectively. To measure performance, we counte the number of times the closest match is a rotated view of the same object. Our result is 285/300. The result reported in [115] is 280/300.

(a) An example objet with all azimuths in sub-database1

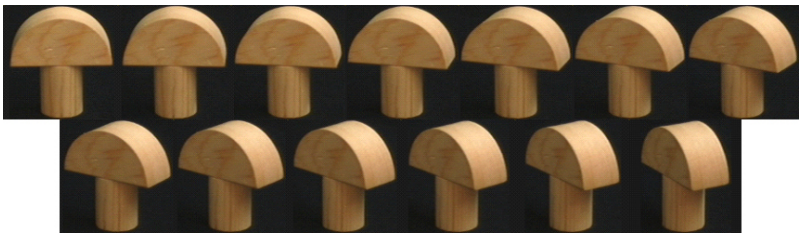


(b) An example objet with all azimuths in sub-database2



(c) An example objet with all azimuths in sub-database3

(d) An example objet with all azimuths in sub-database4



(e) An example objet with all azimuths in sub-database5

(f) An example objet with all azimuths in sub-database6

(g) An example objet with all azimuths in sub-database7

Figure 4.15: The model examples in the 7 sub-databases coming from Coil-100 3D object database

4.2.5.2 Experiment 2

In second experiment, we have generated 7 sub-databases in which we selected 5 to 17 consecutive viewpoints per object in Coil-100 3D object database, for a total from 500 to 1,700 images respectively. We use the middle viewpoint images in a sub-database as a query to retrieve in them. Figure 4.15 shows the model examples in the 7 sub-databases. For each query, the ideal number to retrieve is 5 to 17 in the 7 sub-databases, respectively. The results are shown in the Table 4.5 and Figure 4.16.

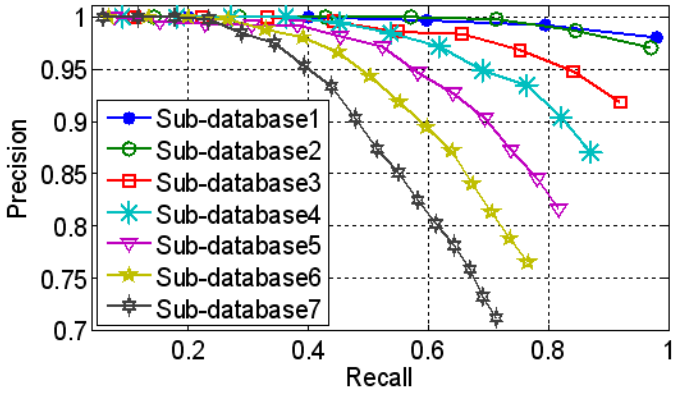


Figure 4.16: Precision and recall rates in the 7 Sub-databases on Coil-100 3D object dataset

These results are very encouraging, since they indicate that we can perform satisfactory retrieval with mean average precision of more than 91% for view angle differences of under 20°: see the results of Sub-database1-3. For viewpoint difference of less than 40°, the retrieved precision is more than 71%: see the results of Sub-database7. Note that this is done exclusively on shape images (without using any intensity information). Clearly, if other information and a more specialized feature set were used, even higher precision scores could be achieved.

4.3 Conclusion

We have presented and evaluated a new approach, the "chord context", for shape recognition problem which is simple and easy to apply. This approach brings two major contributions: 1) defining a new algorithm which can capture the main feature of a shape, from either its contour or a region; 2) proposing an assistant similarity measure algorithm the "Perpendicular Chord Length Eccentricity" (PCLE) which can help to improve retrieval precision and recall to some extent. In our experiments, we have demonstrated invariance to several common image transforms, such as scaling, rotation, boundary perturbations, minor partial occultation and non-rigid deformations. Furthermore, the last evaluation on real-world objects emphasizes the advantages of our approach in the context of machine vision applications.

The main advantages of the proposed descriptor for retrieving images are summarized in the following two points: the flexibility and the accuracy. About flexibility, chord context can handle various types of 2D queries, even if the shape has holes or is composed of several parts. It is robust to noise and minor occultation. So this method doesn't need sophisticate segmentation algorithms to be used. About accuracy, the proposed method has the advantage of achieving higher retrieval accuracy than other methods in the literature evaluated on MPEG-7 CE-1 database, Coil-100 database and the Kimia silhouettes datasets retrieval test.

	Models in Sub-database (azimuth)	Query (azimuth)	The ideal number to retrieve	Retrieved number	Precision
Sub-database1	0, 5, **10,** 15, 20	10	500	490	98.0%
Sub-database2	0, 5, 10, **15,** 20, 25, 30	15	700	679	97.0%
Sub-database3	0, 5, 10, 15, **20,** 25, 30, 35, 40	20	900	827	91.9%
Sub-database4	0, 5, 10, 15, 20, **25,** 30, 35, 40, 45, 50	25	1100	958	87.1%
Sub-database5	0, 5, 10, 15, 20, 25, **30,** 35, 40, 45, 50, 55, 60	30	1300	1062	81.7%
Sub-database6	0, 5, 10, 15, 20, 25, 30, **35,** 40, 45, 50, 55, 60, 65, 70	35	1500	1149	76.6%
Sub-database7	0, 5, 10, 15, 20, 25, 30, 35, **40,** 45, 50, 55, 60, 65, 70, 75, 80	40	1700	1211	71.2%

Table 4.5: The components of 7 sub-databases with different viewpoints on Coil-100 3D object dataset and their retrieval results

Conclusion

In this thesis, we have studied, analyzed, introduced and compared different algorithms for shape-based feature extraction and similarity measure: this work has lead to three approaches for shape-based feature extraction.

For the purposes of understanding the problems and issues involved in shape representations and similarity measures, some fundamental concepts have been introduced. In general, a shape can be represented by its contour, region or finite point set. We have proposed an approach for contour equidistance normalization and discussed some remarkable problems about this normalization. And then we studied some common shape parameters representing certain shape geometric features that can only discriminate shapes with large differences, but are not suitable to be standalone shape descriptors. Some distance functions, operating in a metric distance, that were usually used in similarity measure have been introduced. Since the perception of human does not hold in a metric distance, a simple elastic matching method in non-metric space has also been presented in this part. The evaluation methods for retrieval system performances were defined.

After introducing the ideal properties of the shape features and descriptions, we made a study and a comparison of the methods for shape-based feature extraction and representation. About forty techniques for extraction of shape features have been described and compared. The description of the methods was classified by their processing approaches. These processing approaches include shape signatures, polygonal approximation methods, spatial interrelation feature, moments approaches, scale-space methods and shape transform domains: in such way, one can easily select the appropriate processing approach. To summarize this description, a synthetic table comparing their performances has also been established. Extracting a shape feature in accordance with human perception is not an easy task. Due to the fact that human vision and perception are an extraordinary complicated system, it is a utopia to hope that the machine vision has super excellent performance with small complexity. In addition, choosing appropriate features for a shape recognition system must consider what kinds of features are suitable for the task.

We have proposed two approaches for affine invariant shape descriptions. Both descriptors are based on iso-area normalization (IAN). With inspiration from psychophysics, we proposed the first descriptor, viz., scale-controlled area-difference shape descriptor (SCAD). The proposed descriptor is a vector with components issued for each segment and the ratio of the corresponding area and its filtered segment. This method has three major contributions:

1. it compares the same segment under different scales representation;

2. it chooses the appropriate scales by applying a threshold to the shape shortest-segment;

3. it proposes the algorithm and the conditions of merging and removing the short-segments.

We then defined the second affine invariant descriptor, NPAV. We proved and demonstrated, by two proposed theorems, that for any linearly filtered contour, the area of a triangle, whose vertices respectively are the centroid of the contour and a pair of successive points on the normalized contour, remains linear under affine transforms. So the relation between a filtering and affine transformation is set up. In theory, NPAV remains absolutely invariant under affine transformations. Experimental results indicate that the proposed method is unaffected by boundary starting point variations, affine transforms and serious noise contamination on the shapes. Comparing the approaches SCAD and NPAV, we found that the SCAD descriptor is more compact than the NPAV descriptor but the performance of NPAV is much better than SCAD.

The other proposed approach is the "chord context" shape descriptor. This approach makes the statistics of the lengths of all parallel chords in a shape, with the same interval, and it builds a histogram of their lengths in each direction. This method has two major contributions:

1. definition of a new algorithm which can capture the main feature of a shape, from either its contour or a region;

2. proposition of an assistant similarity measure algorithm Perpendicular Chord Length Eccentricity (PCLE) which can help to improve retrieval precision and recall to some extent.

The main strengths of the proposed descriptor are flexibility and accuracy. Experimental results demonstrate that chord context shape representation is invariant with several common images transforms, such as scaling, rotation, boundary perturbations, minor partial occultation and non-rigid deformations. Furthermore, an evaluation on real-world objects emphasizes the advantages of this approach in the context of machine vision applications.

As a future work, we aim, on one hand, to benefit from the multi-scale and statistic aspects of our approaches to extract the attributes and, on the other hand, to apply elastic matching for similarity measurement. Indeed, the multi-scale analysis is consistent with visual perception and statistic methods have good advantage to similarity measures; in other way, the elastic matching has an interesting property for finding optimal alignment between two shape descriptors. Considering that NPAV has abundant patterns at multiple scales, it is well-adapted for multi-scale analysis. From the other point of view, the chord context method is a statistical approach and it offers excellent performances on similarity-based measures, but unfortunately, it has no special operations that resist affine transforms, so the challenge will be to find affine-invariant features. With the previous considerations, once that these two tools will have been improved, they will be combined jointly with elastic matching to provide an efficient shape-based pattern recognition tool.

Appendix

A. Distance transform (DT)

Distance transform is such an operation which measures the distance of non-edge pixels to the nearest edge pixel while the edge pixels get the value zero. The purpose of the distance transformation is to produce numeric image whose pixels are labeled with distance between each of them and their closet border pixel. It is important that the DT used in the matching algorithm is a reasonably good approximation of the Euclidean distance. One approximation of the Euclidean distance is the chamfer metric [86, 116]. The chamfer 3/4 distance can be calculated sequentially by a two-pass algorithm. First, a distance image is created such that each boundary pixel is set to zero and each non boundary pixel is set to infinity. The forward pass modifies the distance image as follows:

for i=2, 3, $\cdots$, rows do
  for j=2, 3, $\cdots$, columns do
    $v_{i,j} = \min(v_{i-1,j-1} + 4,\ v_{i-1,j} + 3,\ v_{i-1,j+1} + 4,\ v_{i,j-1} + 3,\ v_{i,j})$

Similarly, the backward pass operates as follows:

for i=rows-1, rows-2, $\cdots$, 1 do
  for j=columns-1, columns-2, $\cdots$, 1 do
    $v_{i,j} = \min(v_{i+1,j+1} + 4,\ v_{i+1,j} + 3,\ v_{i+1,j-1} + 4,\ v_{i,j+1} + 3,\ v_{i,j})$

where the $v_{i,j}$ is the value of the pixel in position $(i,\ j)$.

One example of edge map and distance map is illustrated in Figure 17.

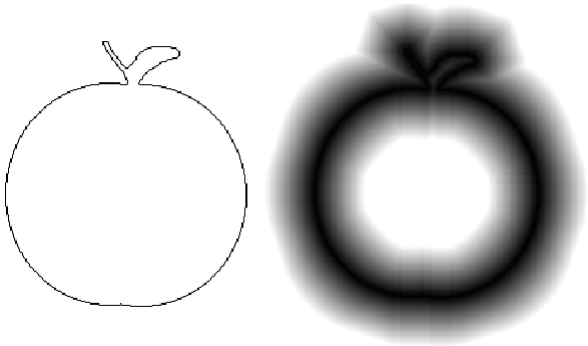


Figure 17: Examples of chamfer 3/4 distance image

B. Contour noise simulation

The general method for noise simulation on a contour is to shift the $x,\ y$ coordinates of the points on a contour independently. In this way, the order of some points on the contour probably becomes confused, causing self-intersection (cf. Figure 18).

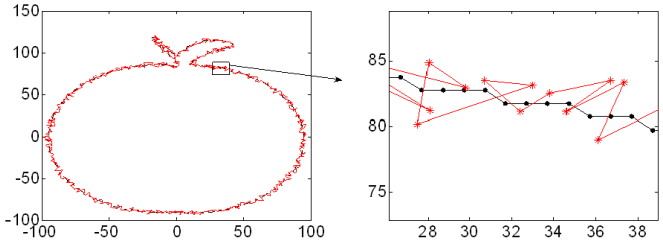


Figure 18: The general method effects of simulated additional noise

But in practice, this distortion does not exist. In order to simulate the real effect of noise on an object, we design the algorithm of adding noise to the test contour as follows.

To decrease or avoid bringing non-real distortion as Figure 18, we control the displacement direction of each point affected by noise to be orthogonal to the tangent of the point. Figure 19. The indication of the displacement direction of a point in the process of simulating noise.

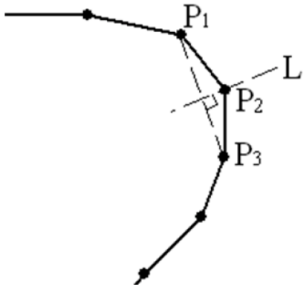


Figure 19: The indication of the displacement direction of a point in the process of simulating noise

See Figure 19: P_1, P_2 and P_3 are the consecutive points on the contour. For simplicity's sake, let us suppose that the direction of the tangent of point P_2 is the direction of the line P_1P_3. The displacement of point P_2 affected by noise is along line L which is orthogonal to the line P_1P_3. Amplitude is controlled by a uniform random value. Its range is [-r, r]. Suppose the average distance of all the points on the contour to its centroid is R, we define the signal-to-noise ratio (SNR) as follows:

$$SNR = 20\log\frac{2R}{r}(dB)$$

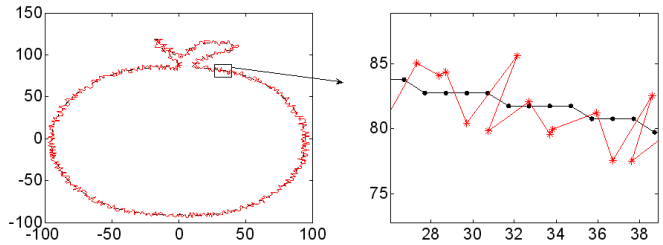


Figure 20: The general method effects of simulated additional noise

Figure 20 visualize the effect of noise simulation on the proposed method. Notice the self-intersection decreasing greatly.

C. Shape Database

C.1 Database of Kimia silhouettes with 216 shapes

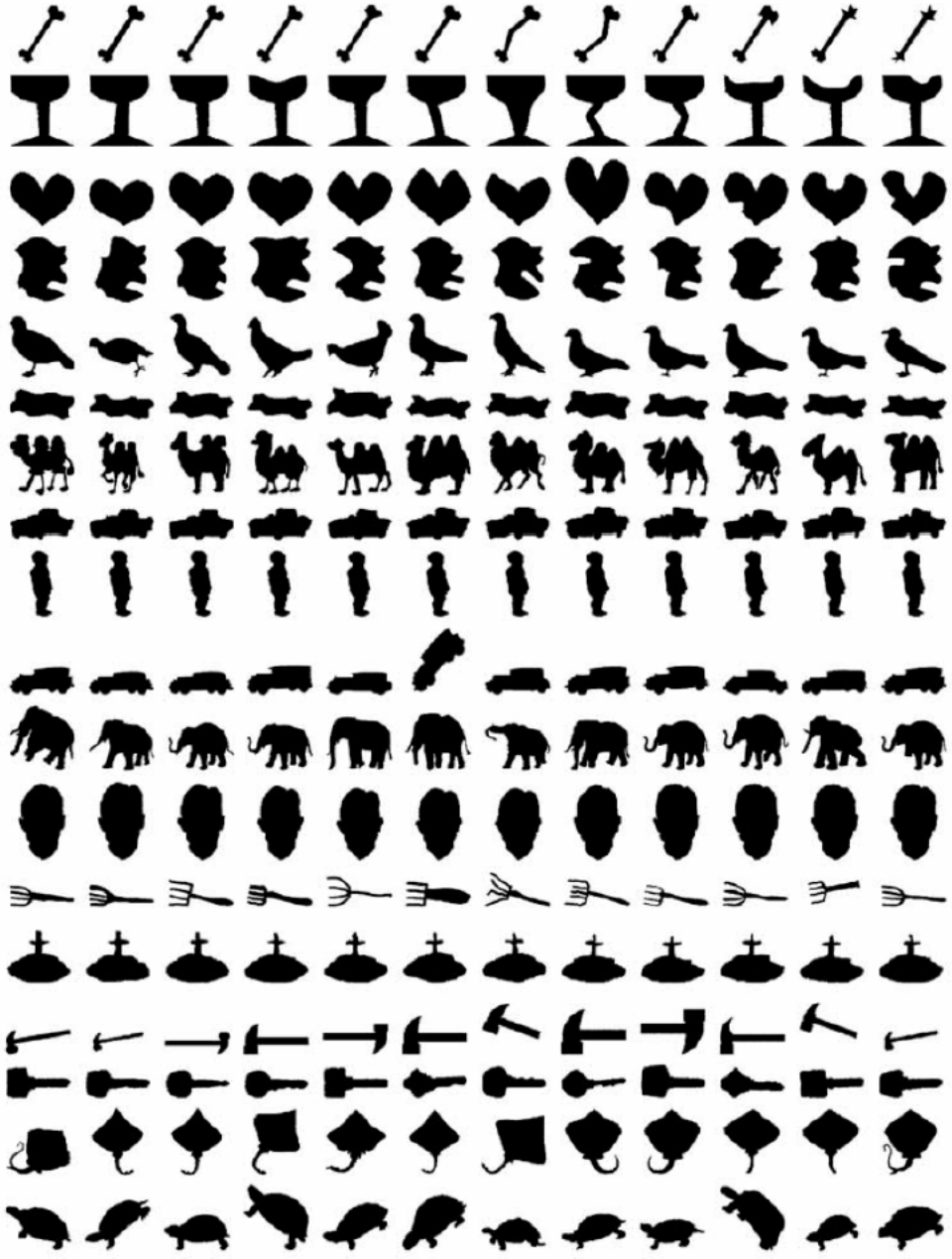


Figure 21: Database of Kimia silhouettes with 216 shapes

C.2 Database of Kimia silhouettes with 99 shapes

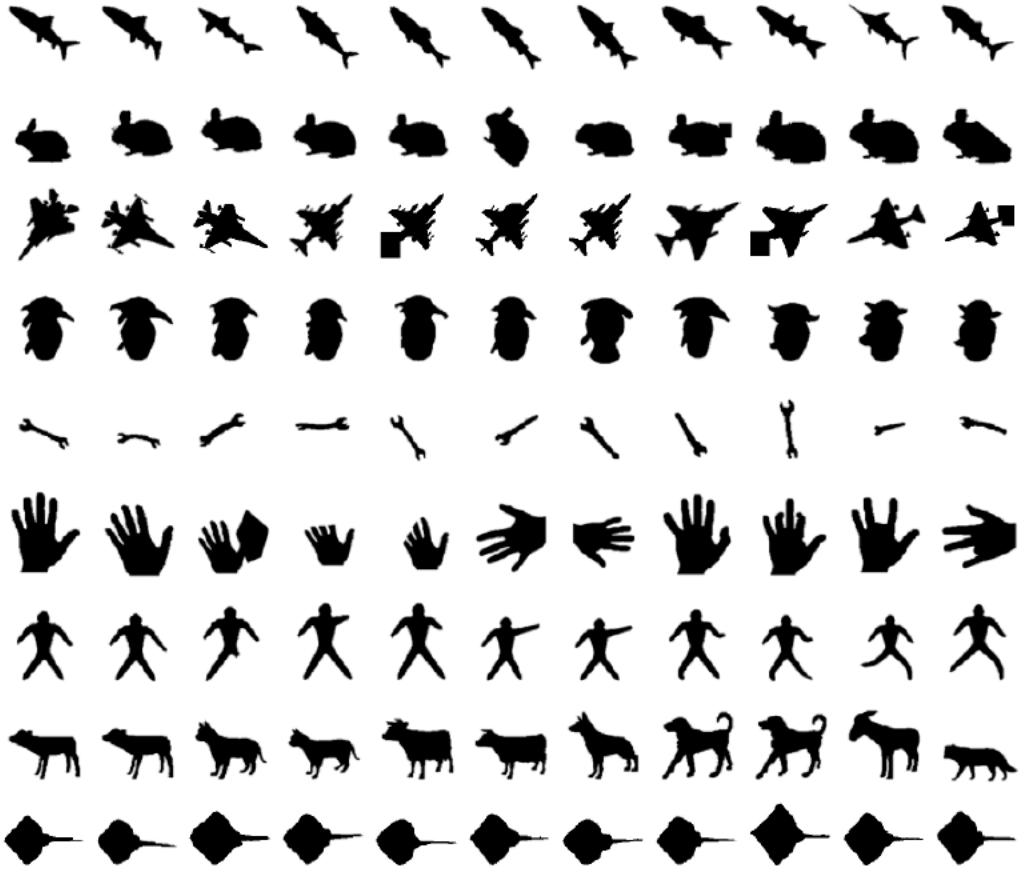


Figure 22: Database of Kimia silhouettes with 99 shapes

C.3 Database of Kimia silhouettes with 25 shapes

Figure 23: Database of Kimia silhouettes with 25 shapes

List of Figures

List of Tables

List of published/accepted papers

The contents of this thesis represent the culmination of a collection of papers published over the span of my thesis. The publications drawn upon are:

Journal papers:

[1] Yang Mingqiang, Kpalma Kidiyo, and Ronsin Joseph, "Affine invariance contour descriptor based on iso-area normalization," IEE, Electronics Letters, Vol.43(7), pp.379-380, Mar. 2007.

[2] Yang Mingqiang, Kpalma Kidiyo, and Ronsin Joseph, "Affine invariance contour descriptor based on the equal area normalization," IAENG International Journal of Applied Mathematics, Vol.36(2), pp. 27-32. Jun. 2007.

Conference papers:

[3] Yang Mingqiang, Kpalma Kidiyo, and Ronsin Joseph, "Scale-controlled area difference shape descriptor," SPIE Vol.6500, Proceedings Electronic Imaging science and Technology, 2007.

[4] Yang Mingqiang, Kpalma Kidiyo, and Ronsin Joseph, "An affine invariant contour descriptor based on filtered enclosed area," Proceedings the International MultiConference of Engineers and Computer Scientists 2007, IMECS 2007, Mar. 21-23, Hong Kong, China. pp. 1907-1912.

[5] Yang Mingqiang, Kpalma Kidiyo, and Ronsin Joseph, "Shape matching and object recognition using chord contexts," 12th International Conference Information Visualisation (IV08), 8-11 July 2008.

[6] Kpalma Kidiyo, Yang Mingqiang, and Ronsin Joseph, "Planar shapes descriptors based on the turning angle scalogram," International Conference on Image Analysis and Recognition (ICIAR 2008).

Papers in books as a chapter:

[7] Yang Mingqiang, Kpalma Kidiyo, and Ronsin Joseph, "Contour descriptor based on affine invariance normalization," Advances in Communication Systems and Electrical Engineering. February 2008. ISBN: 978-0-387-74937-2.

List of submitted papers

[1] Yang Mingqiang, Kpalma Kidiyo, and Ronsin Joseph, "Chord context descriptor for effective shape retrieval," IEEE Trans. on Image processing.

[2] Yang Mingqiang, Kpalma Kidiyo, and Ronsin Joseph, "A survey of shape feature extraction techniques," Pattern Recognition; Techniques, Technology and Applications.

Bibliography

[1] T. Sikora, "The MPEG-7 visual standard for content description-an overview," *IEEE Tran. Circuits and Systems for Video Technology*, vol. 11(6), pp. 696–702, 2001.

[2] "MPEG-7 overview (version 10)," ISO/IEC JTC1/SC29/WG11, Tech. Rep., 2004.

[3] K. KPALMA and J. RONSIN, *Vision systems segmentation and pattern recognition.* I-Tech Education and Publishing, 2007, ch. An overview of advances of pattern, pp. 169–194.

[4] S. Loncaric, "A survey of shape analysis techniques," *Pattern Recognition*, vol. 31(8), pp. 983–1001, 1998.

[5] S. Berretti, A. D. Bimbo, and P. Pala, "Retrieval by shape similarity with perceptual distance and effective indexing," *IEEE Trans. on Multimedia*, vol. 2(4), pp. 225–239, 2000.

[6] M. Tarr and H. Bulthoff, "Image-based object recognition in man, monkey, and machine," *Cognition*, vol. 67, pp. 1–20, 1998.

[7] R. Veltkamp and M. Hagedoorn, "State of the art in shape matching," *Principles of Visual Information Retrieval*, pp. 89–119, 2001.

[8] D. Zhang and G. Lu, "Review of shape representation and description techniques," *Pattern Recognition*, vol. 37, pp. 1–19, 2004.

[9] B. A. Olshausen, "A survey of visual preprocessing and shape representation techniques," Research Institute for Advanced Computer Science NASA Ames Kzsearch Center, Tech. Rep., 1988.

[10] M. E. Celebi and Y. A. Aslandogan, "A comparative study of three moment-based shape descriptors," in *Proc. of the International Conference of Information Technology: Coding and Computing*, 2005, pp. 788–793.

[11] E. Persoon and K.-S. Fu, "Shape discrimination using fourier descriptors," *IEEE Trans. Systems, Man and Cybernetics*, vol. SMC-7(3), pp. 170–179, 1977.

[12] D. Zhang and G. Lu, "Shape retrieval using fourier descriptors," in *Proc. the International Conference on Multimedia and Distance Education*, 2001, pp. 1–9.

[13] H. Kauppinen, T. Seppanen, and M. Pietikainen, "An experimental comparison of auto-regressive and fourier-based descriptors in 2-D shape classification," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17(2), pp. 201–207, 1995.

[14] K. Muller and J.-R. Ohm, "Contour description using wavelets," in *Proc. Workshop on Image Analysis for Multimedia Interactive Services*, 1999, pp. 77–80.

[15] C.-H. Chuang and C.-C. Kuo, "Wavelet descriptor of planar curves: Theory and applications," *IEEE Trans. Image Processing*, vol. 5(1), pp. 56–70, 1996.

[16] E. Bala and A. E. Cetin, "Computationally efficient wavelet affine invariant functions for shape recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26(8), pp. 1095–1099, 2004.

[17] F. Mokhtarian, S. Abbasi, and J.Kittler, "Robust and efficient shape indexing through curvature scale space," in *Proc. the British Machine Vision Conference*, 1996, pp. 53–62.

[18] B. Ye and J. xiong Peng, "Improvement and invariance analysis of zernike moments using as a region-based shape descriptor," in *Proc. XV Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI)*, 2002, pp. 419–428.

[19] G. J. Lu and A. Sajjanhar, "Region-based shape representation and similarity measure suitable for content-based image retrieval," *Multimedia Systems*, vol. 7(2), pp. 165–174, 1999.

[20] A. Taza and C. Suen, "Discrimination of planar shapes using shape matrices," *IEEE Trans. System, Man, and Cybernetics*, vol. 19(5), pp. 1281–1289, 1989.

[21] J. Flusser, "Invariant shape description and measure of object similarity," in *Proc. 4th International Conference on Image Processing and its Applications*, 1992, pp. 139–142.

[22] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape context," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24(4), pp. 509–522, 2002.

[23] B. Horn, *Robot Vision*. MIT Press, Cambridge, 1986.

[24] D.-M. Tsai and M. fong Chen, "Object recognition by a linear weight classifier," *Pattern Recognition Letters*, vol. 16, pp. 591–600, 1995.

[25] I. Young, J. Walker, and J. Bowie, "An analysis technique for biological shape," *Comput. Graphics Image Processing*, vol. 25, pp. 357–370, 1974.

[26] M. Peura and J. Iivarinen, "Efficiency of simple shape descriptors," in *Proc. 3rd International Workshop on Visual Form (IWVF3)*, May 1997.

[27] C. Cheng, W. Liu, and H. Zhang, "Image retrieval based on region shape similarity," in *Proc. 13th SPIE symposium on Electronic Imaging, Storage and Retrieval for Image and Video Databases*, 2001.

[28] A. SOFFER, "Negative shape features for image databases consisting of geographic symbols," in *Proc. 3rd International Workshop on Visual Form*, May 1997.

[29] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age," Penn State University, Tech. Rep., 2006.

[30] R. C. Veltkamp and M. Hagedoorn, "State-of-the-art in shape matching," Utrecht University, Tech. Rep., 1999.

[31] T. Skopal, "On fast non-metric similarity search by metric access methods," in *Proc. 10th International Conference on Extending Database Technology (EDBT'06)*, 2006, pp. 718–736.

[32] J. Ramon, *Image analysis and recognition.* Springer, 2004, ch. Finding significant points for a handwritten classification task, pp. 440–460.

[33] R. C. Veltkamp and M. Hagedoorn, "Shape similarity measures, properties and constructions," in *Proc. Advances in Visual Information Systems, Proceedings of the 4th International Conference, VISUAL*, 2000, pp. 467–476.

[34] T. Skopal, "Unified framework for fast exact and approximate search in dissimilarity spaces," *ACM Transactions on Database Systems*, vol. 32(4), p. 29, 2007.

[35] B. Scassellati, S. Alexopoulos, and M. Flickner, "Retrieving images by-2D shape: a comparison of computation methods with human perceptual judgments," in *Proc. SPIE Storage and Retrieval for Image and Video databases*, 1994.

[36] A. D. Bimbo and P. Pala, "Visual image retrieval by elastic matching of user sketches," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19(2), pp. 121–132, 1997.

[37] S. Krishnamachari and M. A. Mottaleb, "Compact color descriptor for fast image and video segment retrieval," in *Proc. SPIE Storage and Retrieval for Media Databases*, 2000.

[38] D. Zhang and G. Lu, "A comparative study of fourier descriptors for shape representation and retrieval," in *Proc. 5th Asian Conference on Computer Vision*, 2002.

[39] R. B. Yadava, N. K. Nishchala, A. K. Gupta, and V. K. Rastogi, "Retrieval and classification of shape-based objects using fourier, generic fourier, and wavelet-fourier descriptors technique: A comparative study," *Optics and Lasers in Engineering*, vol. 45(6), pp. 695–708, 2007.

[40] D. S. Zhang and G. Lu, "A comparative study on shape retrieval using fourier descriptors with different shape signatures," in *Proc. International Conference on Intelligent Multimedia and Distance Education (ICIMADE01)*, 2001.

[41] C. T. Zahn and R. Z. Roskies., "Fourier descriptors for plane closed curves," *IEEE Trans. Computer*, vol. c-21(3), pp. 269–281, 1972.

[42] K.-J. Lu and S. Kota, "Compliant mechanism synthesis for shape-change applications: Preliminary results," in *Proceedings of SPIE Modeling, Signal Processing, and Control Conference*, vol. 4693, March 2002, pp. 161–172.

[43] L. J. Latecki and R. Lakamper, "Shape similarity measure based on correspondence of visual parts," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22(10), pp. 1185–1190, 2000.

[44] Y. P. Wang, K. T. Lee, and K.Toraichi, "Multiscale curvature-based shape representation using B-spline wavelets," *IEEE Trans. Image Process*, vol. 8(10), pp. 1586–1592, 1999.

[45] F. Mokhtarian and A. K. Mackworth, "A theory of multiscale, curvature-based shape representation for planar curves," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14(8), pp. 789–805, 1992.

[46] A. C. Jalba, M. H. F. Wilkinson, and J. B. T. M. Roerdink, "Shape representation and recognition through morphological curvature scale spaces," *IEEE Trans. Image Processing*, vol. 15(2), pp. 331–341, 2006.

[47] N. Alajlan, M. S. Kamel, and G. Freeman, "Multi-object image retrieval based on shape and topology," *Signal Processing: Image Communication*, vol. 21, pp. 904–918, 2006.

[48] N. Alajlan, I. E. Rube, M. S. Kamel, and G. Freeman, "Shape retrieval using triangle-area representation and dynamic space warping," *Pattern Recognition*, vol. 40(7), pp. 1911–1920, 2007.

[49] S. Han and S. Yang, "An invariant feature representation for shape retrieval," in *Proc. Sixth International Conference on Parallel and Distributed Computing, Applications and Technologies*, 2005.

[50] L. J. Latecki and R. Lakamper, "Convexity rule for shape decomposition based on discrete contour evolution," *Computer Vision and Image Understanding*, vol. 73(3), pp. 441–454, 1999.

[51] K. Chakrabarti, M. Binderberger, K. Porkaew, and S. Mehrotra, "Similar shape retrieval in mars," in *Proc. IEEE International Conference on Multimedia and Expo*, 2000.

[52] C. Bauckhage and J. K. Tsotsos, "Bounding box splitting for robust shape classification," in *Proc. IEEE International Conference on Image Processing*, 2005, pp. 478–481.

[53] E. Davies, *Machine Vision: Theory, Algorithms, Practicalities.* Academic Press, New York, 1997.

[54] R. Gonzalez and R. Woods, *Digital image processing, Second Edition.* PEARSON EDUCATION NORTH ASIA LIMITED and Publishing House of Electronics Industry, 2002.

[55] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis and Machine Vision.* Chapman and Hall, London, UK, 1993.

[56] Y. K. Liu, W. Wei, P. J. Wang, and B. Zalik, "Compressed vertex chain codes," *Pattern Recognition*, vol. 40(11), pp. 2908–2913, 2007.

[57] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. Information Theory*, vol. IT-8, pp. 179–187, 1962.

[58] J. Iivarinen and A. Visa, "Shape recognition of irregular objects," in *Proc. SPIE, Intelligent Robots and Computer Vision XV: Algorithms, Techniques, Active Vision, and Materials Handling*, vol. 2904, 1996, pp. 25–32.

[59] D. Guru and H. Nagendraswam, "Symbolic representation of two-dimensional shapes," *Pattern Recognition Letters*, vol. 28, pp. 144–155, 2007.

[60] N. Arica and F. Vural, "BAS: a perceptual shape descriptor based on the beam angle statistics," *Pattern Recognition Letters*, vol. 24(9-10), 2003.

[61] G. Lu and A. Sajjanhar, "Region-based shape representation and similarity measure suitable for content based image retrieval," *ACM Multimedia System Journal*, vol. 7(2), pp. 165–174, 1999.

[62] G. Mori and J. Malik, "Estimating human body configurations using shape context matching," in *Proc. 7th European Conference on Computer Vision*, vol. III, 2002, pp. 666–680.

[63] H. Zhang and J. Malik, "Learning a discriminative classifier using shape context distances," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003.

[64] A. Thayananthan, B. Stenger, P. H. S. Torr, and R. Cipolla, "Shape context and chamfer matching in cluttered scenes," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003.

[65] S. P. Smith and A. K. Jain, "Chord distribution for shape matching," *Computer Graphics and Image Processing*, vol. 20, pp. 259–271, 1982.

[66] K. Siddiqi and B. Kimia, "A shock grammar for recognition," in *Proceedings of the IEEE Conference Computer Vision and Pattern Recognition*, June 1996, pp. 507–513.

[67] T. Sebastian, P. Klein, and B. Kimia, "Recognition of shapes by editing their shock graphs," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26(5), pp. 550–571, 2004.

[68] G. Taubin and D. Cooper, "Recognition and positioning of rigid objects using algebraic moment invariants," in *SPIE Conference on Geometric Methods in Computer Vision*, vol. 1570, 1991, pp. 175–186.

[69] R. Mukundan, S. Ong, and P. Lee, "Image analysis by tchebichef moments," *IEEE Trans. Image Processing*, vol. 10(9), pp. 1357–1364, 2001.

[70] R. Mukundan, "A new class of rotational invariants using discrete orthogonal moments," in *Sixth IASTED International Conference on Signal and Image Processing*, 2004, pp. 80–84.

[71] K. Jin, M. Cao, S. Kong, and Y. Lu, "Homocentric polar-radius moment for shape classification," in *Proc. Signal Processing, The 8th International Conference on*, 2006.

[72] C. Kan and M. D. Srinath, "Invariant character recognition with Zernike and orthogonal Fourier-Mellin moments," *Pattern Recognition*, vol. 35, pp. 143–154, 2002.

[73] B. M. Mehtre, M. S. Kankanhalli, and W. F. Lee, "Shape measures for content based image retrieval: A comparison," *Pattern Recognition*, vol. 33(3), pp. 319–337, 1997.

[74] J. Peng, W. Yang, and Z. Cao, "A symbolic representation for shape retrieval in curvature scale space," in *Proc. International Conference on Computational Inteligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce*, 2006.

[75] D. Zhang and G. Lu, "A comparative study of curvature scale space and fourier descriptors for shape-based image retrieval," *Visual Communication and Image Representation*, vol. 14(1), 2003.

[76] S. Abbasi, F. Mokhtarian, and J. Kittler, "Enhancing css-based shape retrieval for objects with shallow concavities," *Image and Vision Computing*, vol. 18(3), pp. 199–211, 2000.

[77] M. Yang, K. Kpalma, and J. Ronsin, "Scale-controlled area difference shape descriptor," in *Proc. SPIE, Electronic Imaging science and Technology*, vol. 6500, 2007.

[78] K. Kpalma and J. Ronsin, "Multiscale contour description for pattern recognition," *Pattern Recognition Letters*, vol. 27, pp. 1545–1559, 2006.

[79] D. S. Zhang and G. Lu, "Enhanced generic fourier descriptors for object-based image retrieval," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP2002)*, pp. 3668–3671.

[80] G. C.-H. Chuang and C.-C. J. Kuo, "Wavelet descriptor of planar curves: Theory and applications," *IEEE Trans. Image Processing*, vol. 5(1), pp. 56–70, 1996.

[81] M. Khalil and M. Bayoumi, "A dyadic wavelet affine invariant function for 2D shape recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25(10), pp. 1152–1164, 2001.

[82] G. Chen and T. D. Bui, "Invariant fourier-wavelet descriptor for pattern recognition," *Pattern Recognition*, vol. 32, pp. 1083–1088, 1999.

[83] J. Ricard, D. Coeurjolly, and A. Baskurt, "Generalizations of angular radial transform for 2D and 3D shape retrieval," *Pattern Recognition Letters*, vol. 26(14), 2005.

[84] S.-M. Lee, A. L. Abbott, N. A. Clark, and P. A. Araman, "A shape representation for planar curves by shape signature harmonic embedding," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.

[85] S. Tabbone, L. Wendling, and J.-P. Salmon, "A new shape descriptor defined on the radon transform," *Computer Vision and Image Understanding*, vol. 102(1), pp. 42–51, 2006.

[86] G. Borgefors, "Distance transformations in digital images," in *Computer Vision, Graphics, and Image Processing*, June 1986, pp. 344–371.

[87] G. S. di Baja and E. Thiel, "Skeletonization algorithm running on path-based distance maps," *Image Vision Computer*, vol. 14, pp. 47–57, 1996.

[88] A. Dubinskiy and S. C. Zhu, "A multi-scale generative model for animate shapes and parts," in *Proc. Ninth IEEE International Conference on Computer Vision (ICCV)*, 2003.

[89] K. Arbter, W. Snyder, H. Burkhardt, and G. Hirzinger, "Applications of affine-invariant fourier descriptors to recognition of 3-D objects," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12(7), pp. 640–646, 1990.

[90] H. Guggenheimer, *Differential Geometry*. McGraw Hill, 1963.

[91] Q. M. Tieng and W. W. Boles, "Wavelet-based affine invariant representation: A tool for recognizing planar objects in 3D space," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19(8), pp. 846–857, 1997.

[92] T. Reiss, "The revised fundamental theorem of moment invariants," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13(8), pp. 830–834, 1991.

[93] F. Bashir and A. Khokhar, "Curvature scale space based affine-invariant trajectory retrieval," in *Proc. 8th International Multitopic Conference*, 2004, pp. 24–26.

[94] F. Mokhtarian and S. Abbasi, "Shape similarity retrieval under affine transform," *Pattern Recognition*, vol. 35(1), pp. 31–41, 2002.

[95] D. Cyganski and R. Vaz, "A linear signal decomposition approach to affine invariant contour identification," in *Proce. SPIE Intelligent Robots and Computer Vision X: Algorithms and Techniques*, 1991, pp. 98–109.

[96] H. Sahbi, "Affine invariant shape description using the triangular kernel and its application to leaf recognition," in *Proc. 4th International Workshop on Content-Based Multimedia Indexing*, 2005.

[97] A. Ghosh and N. Petkov, "Robustness of shape descriptors to incomplete contour representations," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27(11), pp. 1793–1804, 2005.

[98] L. J. Latecki, R. Lakamper, and U. Eckhardt, "Shape descriptors for non-rigid shapes with a single closed contour," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2000, pp. 424–429.

[99] F. Mokhtarian, "Silhouette-based isolated object recognition through curvature scale," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, pp. 539–544, 1995.

[100] N. Ueda and S. Suzuki, "Learning visual models from shape contours using multiscale convex/concave structure matching," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15(4), pp. 337–352, 1993.

[101] I. Kunttu, "Shape and gray level descriptors for surface defect image retrieval and classification," Ph.D. dissertation, Tampere University of Technology, 2005.

[102] B. Kimia and K. Siddiqi, "Geometric heat equation and nonlinear diffusion of shapes and images," *Compute Vision Image Understanding*, vol. 64(3), pp. 305–332, 1996.

[103] M. Yang, K. Kpalma, and J. Ronsin, "Affine invariance contour descriptor based on iso-area normalization," *Electronics Letters*, vol. 43(7), pp. 379–380, 2007.

[104] M. B. S. Jeannin, "Description of core experiments for mpeg-7 motion/shape," in *MPEG-7, ISO/IEC JTC1/SC29/WG11/MPEG99/N2690*, March 1999.

[105] F. Mokhtarian, S. Abbasi, and J. Kittler, "Efficient and robust retrieval by shape content through curvature scale space," in *In image databases and multi media search, proceeding of the first international workshop IDB-MMS'96*, 1996, pp. 35–42.

[106] O. E. Badawy and M. Kamel, "Shape retrieval using concavity trees," in *Proceedings of the 17th International Conference on Pattern Recognition*, 2004, pp. 111–114.

[107] D. Sharvit, J. Chan, H. Tek, and B. Kimia, "Symmetry-based indexing of image databases," in *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries*, 1998.

[108] M. Yang, K. Kpalma, and J. Ronsin, "Shape matching and object recognition using chord contexts," in *12th International Conference Information Visualisation (IV08)*, July 2008.

[109] T. Bernier and J.-A. Landry, "A new method for representing and matching shapes of natural objects," *Pattern Recognition*, vol. 36(8), pp. 1711–1723, 2003.

[110] D. available on the website, "http://www1.cs.columbia.edu/cave/software/softlib/coil-100.php."

[111] A. Khotanzad and Y. Hong, "Invariant image recognition by Zernike moments," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12 (5), pp. 289–497, 1990.

[112] Y. Gdalyahu and D. Weinshall, "Flexible syntactic matching of curves and its application to automatic hierarchical classification of silhouettes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21(12), pp. 1312–1328, 1999.

[113] H. Ling and D. Jacobs, "Shape classification using the inner-distance," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29(2), pp. 286–299, 2007.

[114] E. Petrakis, A. Diplaros, and E. Milios, "Matching and retrieval of distorted and occluded shapes using dynamic programming," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24(11), pp. 1501–1516, 2002.

[115] S. Belongie and J. Malik, "Matching with shape contexts," in *IEEE Workshop on Content-Based Access of Image and Video Libraries*, 2000.

[116] H. C. Liu and M. D. Srinath, "Partial shape classification using contour matching in distance transformation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12(11), pp. 1072–1079, 1990.

[117] J. Chen, P. Tan, and T. Goh, "Refinement to the chamfer matching for a center on fit," in *International Conference on Image and Vision Computing (New Zealand)*, 2003, pp. 360–365.

Résumé

Dans le contexte de la reconnaissance de forme et de l'observation de similarité d'un objet parmi d'autres, les caractéristiques de forme extraites de son image sont des outils puissants. En effet la forme de l'objet est habituellement et fortement liée à sa fonctionnalité et son identité. S'appuyant sur cette forme, un éventail de méthodes par extraction de caractéristiques et mesures de similarité a été proposé dans la littérature. De nombreuses et diverses applications sont susceptibles d'utiliser ces caractéristiques de forme. L'invariance géométrique et la résistance aux déformations sont des propriétés importantes que doivent posséder ces caractéristiques et mesures de similarité.

Dans cette thèse, trois nouveaux descripteurs de forme sont développés. Les deux premiers, celui par différence de surfaces et contrôlée par l'échelle (SCAD) et celui correspondant au vecteur de surfaces partielles normalisées (NPAV), sont fondés sur une normalisation "iso-surface" (IAN). SCAD est un vecteur dont les éléments sont les différences de surface entre les principaux segments du contour original et contour filtré. Ces segments sont définis par des ensembles de points entre chaque paire de points de courbure nulle, relative au contour filtré et au contour original. En nous appuyant sur deux théorèmes que nous proposons et en prenant en considération surface partielle, transformée affine et filtrage linéaire, nous avons défini le second descripteur, NPAV. Nous prouvons alors, que pour tout contour filtré linéairement, la surface d'un triangle, dont les sommets sont le barycentre du contour et une paire de points successifs sur le contour normalisé, reste linéaire sous toutes les transformations affines. Ainsi est établie une relation entre filtrage et transformation affine. Les deux descripteurs SCAD et NPAV ont la propriété d'invariance aux transformations affines. Comparant les deux approches SCAD et NPAV, SCAD s'avère plus compact que NPAV mais les performances de NPAV sont meilleures que celles de SCAD. La dernière approche proposée est la représentation par "contexte des cordes". Cette représentation décrit une distribution des longueurs de cordes selon une orientation. L'histogramme représentant ce contexte des cordes est compacté et normalisé dans une matrice caractéristique. Une mesure de similarité est alors définie sur cette matrice. La méthode proposée est insensible à la translation, à la rotation et au changement d'échelle; de plus, elle s'avère robuste aux faibles occultations, aux déformations élastiques et au bruit. En outre, son évaluation sur des objets réels souligne tous ses atouts dans le contexte des applications de la vision.

Ces nouveaux descripteurs de forme proposés sont issus d'une recherche et d'études menées sur une quarantaine de techniques de la littérature. Contrairement à la classification traditionnelle, ici, les approches de descripteurs de forme sont classées selon leurs approches de traitement: ceci facilite ainsi le choix du traitement approprié. Une description et une étude de ces approches est ici fournie, et nous reprenons certaines d'entre elles dans une évaluation comparative avec les nôtres et ce sur différentes bases de données

Mots-clefs: Extraction d'attributs, mesures de similarité, descripteur de forme, reconnaissance de formes

Abstract

For object recognition and similarity retrieval, object shape features are powerful because shape is usually strongly linked to object functionality and identity. A wide range of shape-based feature extraction and similarity measures methods are proposed in literature, moreover various applications are likely to use shape features. For shape-based feature extraction and similarity measure, geometric invariance and deformation resistibility are important issues.

This thesis develops three new shape descriptors. The first two descriptors, scale-controlled area-difference shape descriptor (SCAD) and normalized partial area vector (NPAV), are based on iso-area normalization (IAN). SCAD is a vector whose elements are the area differences between the main corresponding segments of original contour and its filtered version. These main segments are defined by the set of points between each pair of curvature zero-crossing points of the filter contour for a given scale. Based on two proposed theorems taking into consideration partial areas, affine transforms and low pass filtering, we define the second affine invariant descriptor, NPAV. We then prove that for any linearly filtered contour, the area of a triangle, whose vertices respectively are the centroid of the contour and a pair of successive points on the normalized contour, remains linear under affine transforms. So the relation between a filtering and affine transformation is set up. Both SCAD and NPAV have the good property of affine invariance. Comparing the approaches SCAD and NPAV, the SCAD descriptor is more compact than the NPAV descriptor and the performance of NPAV is much better than SCAD. The other proposed approach is chord context shape representation. Chord context describes a frequency distribution of chord lengths with different orientations. The histogram which represents the chord context is compacted and normalized into a feature matrix. A similarity measure is defined on the basis of this feature matrix according to its characteristics. The proposed method is shown to be unaffected by shape translation, rotation and scaling; at the same time, it is robust to minor occultation, non-rigid deformations, distortions and corruption due to noise. Furthermore, an evaluation on real-world objects emphasizes the advantages of this approach in the context of machine vision applications.

These new shape descriptors are proposed based on the extensive investigation and study of about forty existing techniques. In this thesis, unlike the traditional classification, all these approaches of shape-based feature extraction and representation are classified by their processing approaches: in such way, one can easily select the appropriate processing approach. A description and the study of these approaches are developed, and some of them are used in comparative experiments, on different data bases with ours.

Keywords: Feature extraction, similarity measures, shape descriptor, shape retrieval